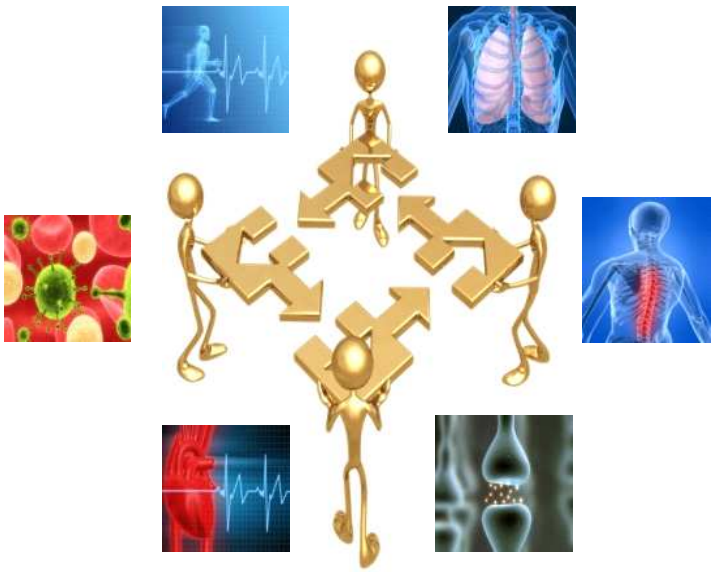
Hes·so VALAIS WALLIS
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

Travail de bachelor 2010

Filière Informatique de gestion

Exchanges of Medical Informations



Etudiant : Arnaud Gaspoz

Professeur : Michael Schumacher

www.hevs.ch

Préface

La cybersanté (eHealth) regroupe tous les services électroniques du domaine de la santé. Les technologies de l'information sont utilisées afin d'améliorer les processus du système de santé et également de mettre en relation les acteurs concernés, tels que les patients, médecins et hôpitaux.

La cybersanté est une jeune discipline qui fait entrer la santé dans l'ère numérique. Actuellement, les milliards de données sont stockées en partie sous forme papier et en partie déjà sous forme électronique. L'échange de données médicales s'effectue par courrier, ce qui est assez lent et peut provoquer des erreurs. Le but de la cybersanté est de créer davantage de qualité dans le domaine de la santé de manière sécurisée et à long terme à contribuer à stabiliser les coûts. [1]

En dépit de son haut niveau de développement, la Suisse accuse un retard par rapport à l'étranger dans le domaine de la cybersanté. Depuis janvier 2006, la Confédération suisse porte un intérêt particulier à la cybersanté et a défini une nouvelle stratégie qui permettrait à un patient d'accéder à ces informations médicales. [2]

Page de garde:
Image 1: MediCoordination.ch

Résumé

Introduction

Le domaine de la santé effectue actuellement le passage d'une méthode basée sur le papier à une méthode électronique. La Confédération suisse vient de publier une nouvelle stratégie dont le but est de numériser les informations médicales, puis de les rendre accessibles immédiatement.

Ce travail de bachelor s'appuie sur le projet MediCoordination.ch dirigé par l'institut Informatique de gestion de Sierre. Un prototype a déjà été implémenté et permet l'échange de données entre hôpitaux et médecins généralistes. Cette implémentation se base sur les profils IHE, ceux-ci étant des recommandations pour rendre les systèmes médicaux de fabricants différents interopérables.

Afin de bien comprendre le contexte, plusieurs sujets ont été abordés, tels que les profils IHE, le projet MediCoordination et son prototype et la stratégie de la Confédération suisse.

Dans un même temps, le responsable de sécurité informatique du RSV a effectué un audit de sécurité du prototype et nous a transmis un rapport contenant les incohérences et recommandations à apporter. Par la suite, certains aspects de la sécurité ont été repris plus en détail par mes soins.

Méthodes

Après avoir lu les recommandations de l'audit de sécurité, une analyse des profils IHE était nécessaire afin de déterminer si certains profils peuvent résoudre certains aspects relevés dans le rapport.

Chaque point de l'audit de sécurité a été repris et des solutions ont été apportées. Par la suite, plusieurs profils IHE sont analysés afin de déterminer la valeur ajoutée à leur implémentation. Les profils étudiés sont BPCP, PIX, XUA, EUA, CT et ATNA.

A la fin de cette analyse, une liste de priorités a été dressée et les profils CT et ATNA ont été retenus pour la suite du travail. Le profil CT (Consistent Time) est employé pour que toutes les machines du réseau aient une horloge synchronisée. Le profil ATNA (Audit Trail and Node Authentication) se décompose en deux parties : la génération de logs et l'authentification mutuelle de toutes les machines du réseau.

Un design reprenant l'architecture du prototype en y incluant les deux nouveaux profils a été établi et est suivi d'une phase d'implémentation. Celle-ci est constituée de la mise ne place d'une base de données pour les logs, d'un serveur de temps pour maintenir l'horloge interne des différentes machines et de la gestion des certificats pour l'authentification mutuelle des acteurs.

Conclusion

Ces deux profils étant implémentés, le prototype bénéficie de la traçabilité médico-légale, ce qui permet à un administrateur de consulter les logs et ainsi prendre les décisions qui conviennent. De plus, chaque nœud est maintenant authentifié comme appartenant au réseau et les communications sont sécurisées.

Table des matières

1 Présentation du travail

1.1 Contexte

Proposé par M. Schumacher, ce travail de bachelor s'appuie sur le projet MediCoordination dont le but est de trouver des solutions concrètes à l'interopérabilité de dossiers électroniques de patients entre les hôpitaux et les acteurs des métiers de la santé.

Le projet MediCoordination[1] (MC) est dirigé par l'institut Informatique de gestion de Sierre. L'équipe MC a implémenté un prototype qui a été soumis à un audit de sécurité effectué par le responsable de la sécurité informatique du RSV de Sion. Celui-ci a transmis un rapport de l'audit en y relevant les incohérences et incompréhensions liées aux documents de référence et au prototype.

L'objectif de ce travail est d'analyser les différents profils IHE qui pourraient répondre aux recommandations proposées dans le rapport. Puis dans un second temps, une sélection d'un ou plusieurs profils sera effectuée et commencera alors une phase de design et d'implémentation.

Ce projet est réalisé dans le cadre d'un travail de bachelor HES en informatique de gestion. Le temps mis à disposition est de 360 heures, à raison de 9 heures par jour.

1.2 Cahier des charges

Afin d'établir les phases du projet, un cahier des charges (voir annexe 15.1) a été rédigé, puis accepté par l'auteur et le responsable du projet lors de la réunion du 10 juin 2010.

Le cahier des charges a permis de définir les phases importantes du projet que sont :

- Comprendre l'architecture actuelle
- Elaborer des audits de sécurité
- Analyser les profils IHE pouvant répondre à certains problèmes de sécurité
- Conceptualiser d'un ou plusieurs aspects décrits dans l'analyse
- Proposer une implémentation

Durant toute la durée du travail, le cahier des charges servira de ligne directrice pour bien suivre les exigences du responsable du projet.

[1] http://www.medicoordination.ch/

2 Comprendre l'architecture actuelle

2.1 Introduction

Le domaine de la santé effectue actuellement le passage d'une méthode basée sur le papier à une méthode électronique. La Confédération suisse vient de publier une nouvelle stratégie dont le but est de numériser les informations médicales, puis de les rendre accessibles immédiatement.

Un prototype a déjà été implémenté par l'équipe MC et permet l'échange de données médicales entre hôpitaux et médecins généralistes. L'implémentation s'est basée sur les profils IHE[2] (Integrating the Healthcare Entreprise), plus spécifiquement sur le profil XDS (Cross Enterprise Document Sharing).

Afin d'avoir une vue globale du projet, plusieurs points sont à éclaircir :

- Les profils IHE
- Le profil XDS
- Le projet MediCoordination
- Le prototype MediCoordination
- Les documents émis par l'organe Cybersanté Suisse

Ces différents concepts sont expliqués ci-dessous.

2.2 Les profils IHE

Integrating the Healthcare Enterprise (IHE) est une initiative développée pour améliorer l'intégration des systèmes d'information dans les institutions modernes de la santé. Son objectif est d'assurer que dans le cas des patients, toutes les informations nécessaires pour une décision médicale sont correctes et disponibles pour les professionnels de la santé. L'initiative IHE définit un cadre technique pour l'implémentation des architectures et ainsi assurer une meilleure interopérabilité entre les systèmes utilisant IHE.

L'approche employée par l'initiative IHE est de supporter l'utilisation de standards existants, tels que HL7[3], DICOM, ISO, OASIS, plutôt que de définir des nouveaux standards [3].

Sur le site officiel d'IHE, il est possible de télécharger un PDF (Portable Document Format) nommé « IHE IT Infrastructure Technical Framework » (ITI TF). Ce document définit des implémentations spécifiques de standards afin de promouvoir le partage d'informations médicales.

[2] http://www.ihe.net/
[3] http://www.hl7.org/

ITI TF offre un langage commun pour que les professionnels de la santé et les vendeurs puissent discuter des besoins des entreprises du domaine de la santé et des capacités d'intégration des systèmes d'information. Les profils d'intégration spécifient les implémentations de standards et sont conçus pour correspondre à des besoins médicaux identifiés [4].

Les profils d'intégration sont définis en termes d'acteurs et de transactions. Les acteurs sont des systèmes d'information ou composants d'un système d'information qui produisent, gèrent et agissent sur les informations liées aux activités de l'entreprise. Les transactions sont les interactions entre les acteurs se traduisant par un ensemble de messages décrit dans des standards, par exemple HL7 et DICOM. [5]

Health Level 7 (HL7) est une organisation qui définit des standards pour le format des données médicales. Ces spécifications tendent à intégrer les normes américaines et internationales (ISO) [6].

Digital Imaging and COmmunications in Medicine (DICOM) est un standard de communication et d'archivage en imagerie médicale. L'objectif du standard DICOM est la généralisation du format de l'imagerie pour faciliter les transferts entre les machines des différents constructeurs [7].

2.3 Le profil XDS

Le profil d'intégration IHE « Cross-Enterprise Document Sharing » (XDS) facilite l'enregistrement, la distribution et l'accessibilité d'archives électroniques de patients à travers les entreprises médicales. Le profil XDS se concentre à fournir des spécifications basées sur des standards pour gérer les échanges de documents entre des entreprises du domaine de la santé.

Le profil XDS.a est obsolète depuis quelques années, le nouveau profil d'intégration se nomme XDS.b. Il est basé sur l'utilisation de Web services et des registres et dépôts ebXML. Pour la suite du document, l'abréviation XDS se réfère à XDS.b.

Le profil XDS suppose que les différentes entreprises qui échangent des documents appartiennent à un domaine d'affinité XDS (XDS Affinity Domain). Ce dernier est un groupe d'entreprises du domaine médical qui ont accepté de travailler ensemble en utilisant un set de règles telles que le format des documents médicaux. La notion de document dans XDS n'est pas limitée à des informations textuelles. Les documents peuvent être des images (DICOM) ou des informations structurées (CDA Release 2) [8].

Clinical Document Architecture (CDA) fournit un modèle d'échange de documents médicaux. CDA consiste en un document structuré, style XML (Extensible Markup Language), avec une section en-tête et corps de document. Il définit la structure d'un document contenant du texte, des images et autres contenus multimédia. CDA fait partie du standard HL7.

L'image 2 donne une vue d'ensemble du profil XDS. Celui-ci est composé d'acteurs et de transactions. La compréhension de ces différents éléments permet d'entrevoir l'implémentation du profil [9].

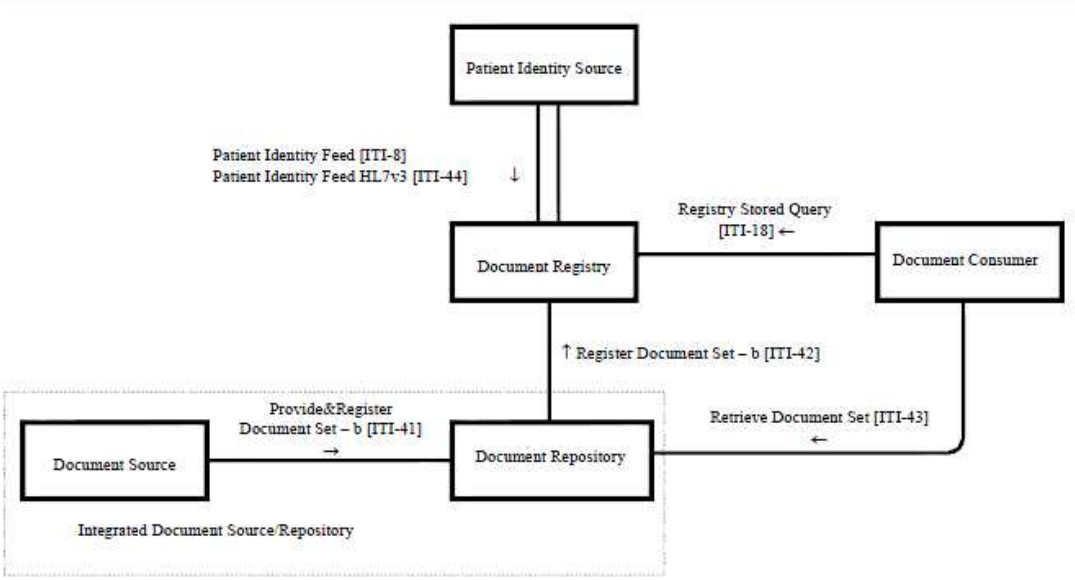


Image 2: Diagramme du profil Cross-Enterprise Document Sharing (XDS)

2.3.1 Acteurs

2.3.1.1 Document Source

L'acteur Document Source est le producteur et l'éditeur des documents. Il est responsable d'envoyer les documents à l'acteur Document Repository qui, par la suite, fournira les métadonnées à l'acteur Document Registry.

2.3.1.2 Document Consumer

Cet acteur interroge l'acteur Document Registry pour des documents correspondant à certains critères, puis récupère les documents sélectionnés auprès de l'acteur Document Repository.

2.3.1.3 Document Registry

L'acteur Document Registry maintient les métadonnées de tous les documents enregistrés, ainsi qu'un lien vers l'endroit où le document est archivé. L'acteur Document Registry répond aux requêtes des acteurs Document Consumer.

2.3.1.4 Document Repository

L'acteur Document Repository est responsable de stocker les documents, ainsi que de les enregistrer auprès de l'acteur Document Registry approprié. Il leur assigne un URI (Uniform Resource Identifier) afin de pouvoir être retrouvés par l'acteur Document Consumer.

2.3.1.5 Patient Identity Source

Cet acteur est un fournisseur d'identifiant unique pour chaque patient. Il facilite la validation des identifiants des patients de l'acteur Registry en interaction avec d'autres acteurs.

Les interactions entre les acteurs sont :

- Provide and Register Document Set
- Register Document Set
- Query Registry
- Registry Stored Query
- Retrieve Document
- Patient Identity Feed
- Retrieve Document Set

Ces transactions ne sont pas expliquées en détail dans ce document, mais sont consultables dans le document ITI TF [4]annexe sur les transactions. Après explications des différents éléments qui constituent le profil XDS, celui-ci devrait être assez clair pour la suite du travail.

2.4 Le projet MediCoordination

Le projet MediCoordination a débuté à l'HES-SO avec pour but de proposer des solutions concrètes à l'interopérabilité des dossiers électroniques des patients entre les hôpitaux et les acteurs de la santé [10]. Il s'agit d'un projet multidisciplinaire qui implique des acteurs de tout bord, que ce soit lors du sondage ou bien lors de l'implémentation. Il y a des acteurs de la HES-SO Santé sociale et Informatique de gestion, des développeurs de solutions médicales et des assurances qui se sont impliqués dans le projet [11]. Cet échange d'informations correspond à un réel besoin médical et financier.

Ce projet suit la tendance actuelle qui vise à rendre interopérable les institutions médicales. Ce fait est observable par la publication de la toute récente stratégie eSanté de la Confédération suisse et quelques années auparavant, une recommandation émise par la Commission européenne.

[4] http://www.ihe.net/Technical_Framework/index.cfm#IT

Le projet MediCoordination a fait un état des lieux de l'échange d'informations entre des entités médicales. Des recherches ont été faites pour connaître les standards, les exigences et spécifications de ces échanges.

Le projet MediCoordination se compose de deux étapes. La première phase correspond à un sondage qui consistait à interroger des acteurs de plusieurs secteurs du domaine de la santé suisse. Grâce aux informations collectées, des diagrammes use-case[5] ont été élaborés en rapport à des scénarios. La seconde phase correspond à l'implémentation du scénario sélectionné [12].

Deux scénarios sont ressortis du sondage :

- La transmission de l'avis de sortie d'un patient de l'hôpital. Ce document résume les informations utiles au médecin de famille, par exemple les radiographies, opérations effectuées durant l'hospitalisation.
- L'admission d'un patient à l'hôpital. Dans ce cas, c'est le médecin de famille qui envoie les informations à l'hôpital [13].

C'est le cas de la transmission de l'avis de sortie d'un patient qui a été retenu pour le prototype. Davantage de détails concernant le prototype sont donnés dans la section suivante.

2.5 Le prototype MediCoordination

Comme dit précédemment, le but technique du prototype est la transmission d'une lettre de sortie de l'hôpital vers le médecin généraliste. La lettre de sortie doit s'intégrer directement dans les archives du médecin généraliste sans intervention manuelle.

L'architecture du prototype MC est basée sur les profils IHE et plus particulièrement le profil XDS. Le prototype permet l'échange de documents. L'hôpital dépose des documents avec les métadonnées correspondantes sur le serveur et le médecin de famille les récupère par la suite.

L'implémentation s'appuie sur plusieurs outils existants. La partie serveur se compose d'iheprofiles d'OpenHealthTools[6] et d'IHE Integration Profile[7] de Microsoft. Pour la partie client, le logiciel MediWay[8] est utilisé. En plus de ces outils, plusieurs composants ont été développés [14].

[5] http://en.wikipedia.org/wiki/Use_case_diagram
[6] http://www.openhealthtools.org/
[7] http://ihe.codeplex.com/
[8] http://www.logival.ch/

2.5.1 OpenHealthTools – iheprofiles

Iheprofiles[9] est une implémentation côté-client de plusieurs profils IHE :

- ATNA (Audit Trail and Node Authentication)
- PDQ (Patient Demographics Query)
- PIX (Patient Identifier Cross-Referencing)
- PAM (Patient Demographics Source)
- XCA (Cross Community Access)
- XDS (Cross-Enterprise Document Sharing)
- XUA (Cross-Enterprise User Assertion)

Ce framework est open source et est utilisé par plus de 35 systèmes au jour d'aujourd'hui [15].

Le bridge OHF est en fait le web service qui sera interrogé par les acteurs Document Source (hôpital) et Document Consumer (médecin généraliste), qui par la suite interrogera le Registry / Repository. Ce composant simplifie les transactions qui peuvent exiger plusieurs appels en un seul message SOAP [16].

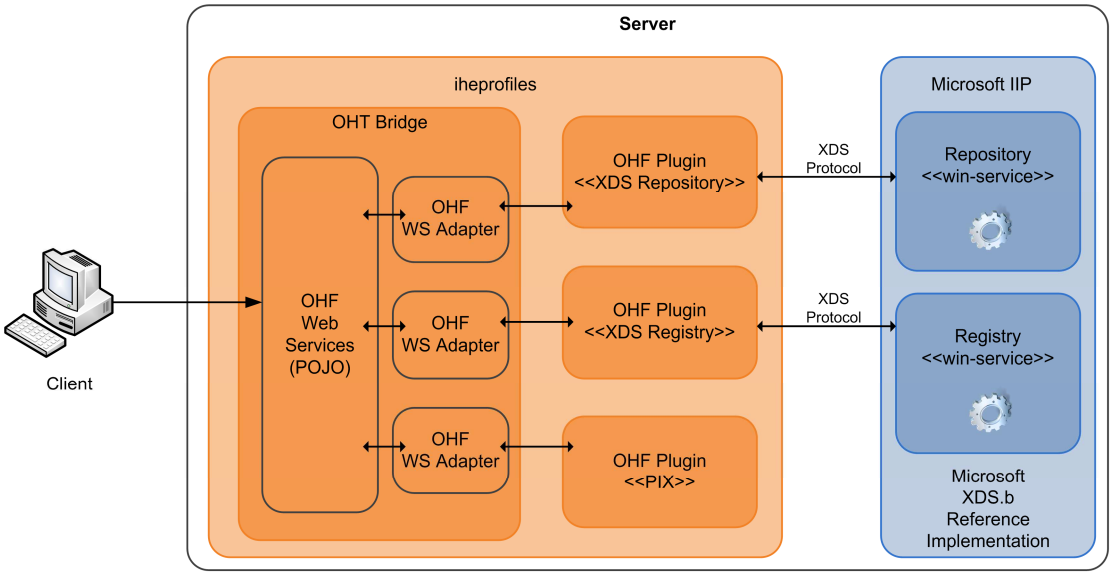


Image 3 : Architecture du bridge OHT

2.5.2 Microsoft – IHE integration Profile

IHE Integration Profile (IIP) est une implémentation côté-serveur du profil XDS. Il implémente les acteurs Registry et Repository, lesquels stockent les métadonnées et les documents. Concrètement, ces deux acteurs sont des services Windows. L'image 4 montre l'implémentation des composants d'IIP [17].

[9] https://iheprofiles.projects.openhealthtools.org/

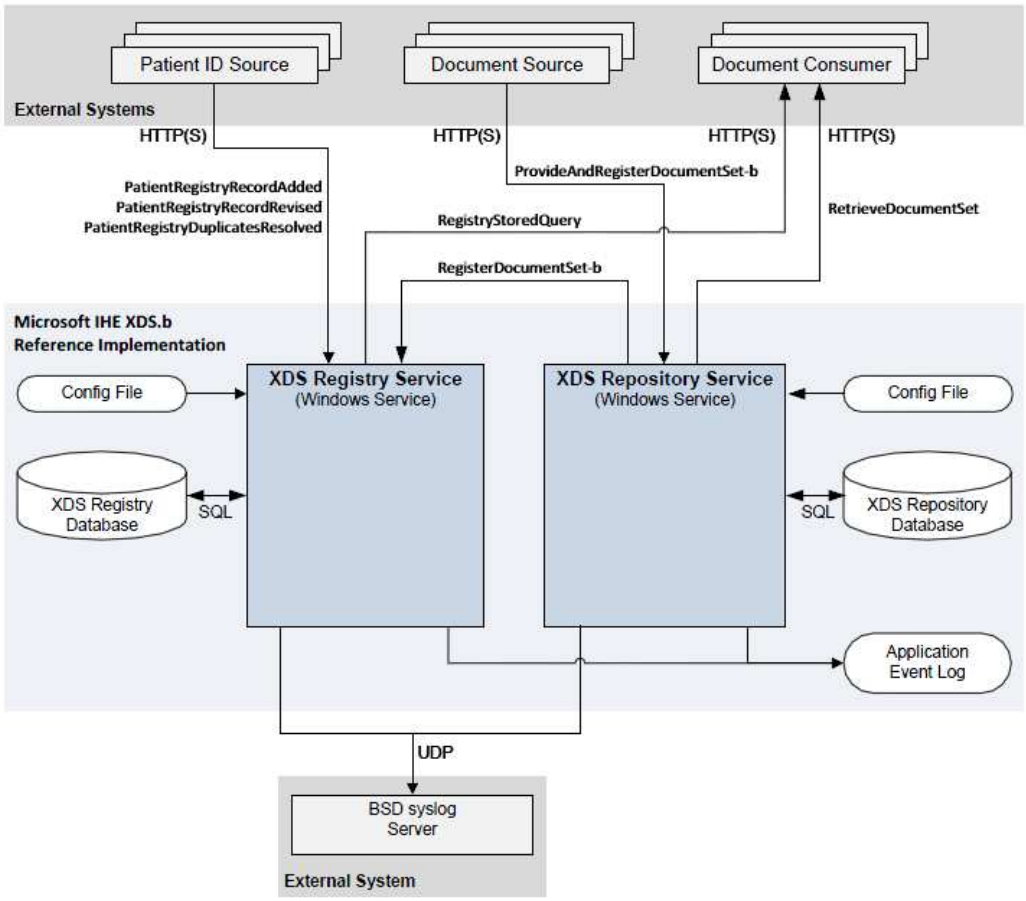


Image 4 : Microsoft IHE XDS.b Reference Implementation

Le nouveau nom de la plateforme est Microsoft IHE XDSb Reference Implementation. Celui-ci est actuellement utilisé dans le cadre du Microsoft Health Information Network (HIN), une solution logicielle proposée par Microsoft.

2.5.3 Vue d'ensemble de l'architecture

L'architecture du prototype est présentée dans l'image 5. La partie centrale représente le serveur où se situent le web service (OHT bridge) et les acteurs Registry / Repository. L'index des patients visible sur le schéma du profil XDS n'est pas implémenté dans le prototype. Sur la gauche se situe l'acteur Document Source qui correspond aux personnels de l'hôpital qui déposent des documents et sur la droite le client, dans notre cas le médecin généraliste qui récupère des documents. Dans notre situation, un seul serveur contient le bridge et les deux services Registry / Repository. Il est également important de noter qu'il n'y a qu'un seul web service dans la solution MediCoordination, contrairement à ce qu'on pourrait croire en regardant l'image 5.

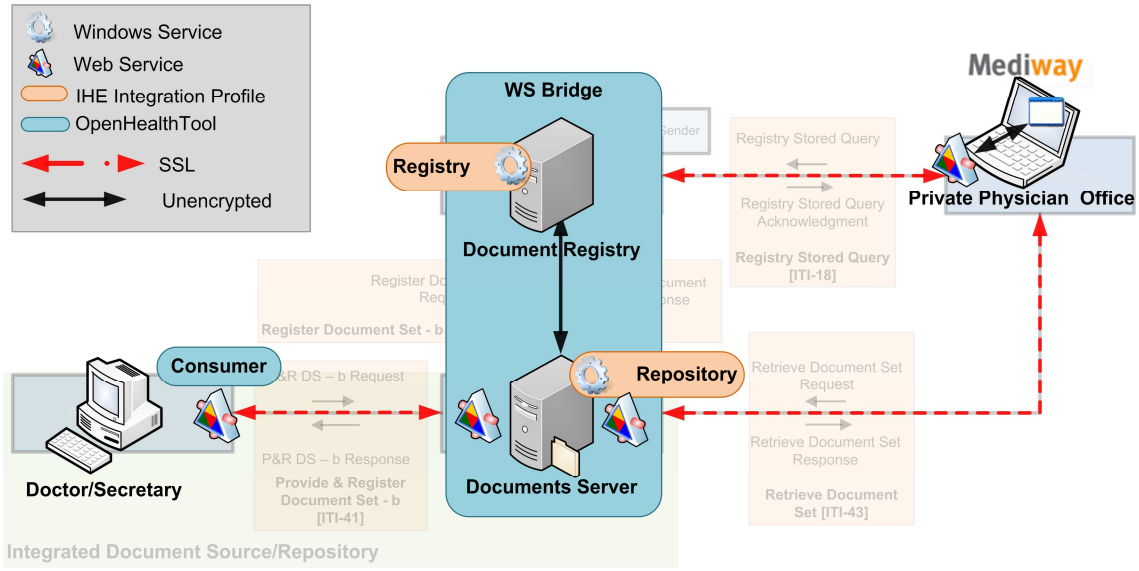


Image 5 : Architecture du prototype MediCoordination

Le prototype est constitué de deux parties : la partie serveur et deux clients.

2.5.4 Partie serveur

La partie serveur est composée des deux outils relevés précédemment que sont iheprofiles d'OpenHealthTools et IHE Integration Profile de Microsoft. IIP implémente le Registry / Repository et iheprofiles est employé pour interroger ces derniers via un web service.

En dehors de ces deux outils, l'équipe MediCoordination a également développé quelques composants appelés ci-après Medico. Medico est une application Java composée de Folder Checker et de MediCo API. Folder Checker vérifie la présence de nouveaux documents dans un dossier défini. Les documents sont transmis via FTP (File Transfer Protocol) dans le dossier cité précédemment. Il transmet ensuite les documents au MediCo API qui se charge de générer un fichier CDA avec le PDF embarqué dans celui-ci. Enfin, les documents sont envoyés via messages SOAP (Simple Object Access Protocol) vers l'OHT Bridge. Le but de l'application Java est de permettre à l'hôpital d'envoyer des documents sans devoir les exporter au format CDA.

Lorsque le Bridge reçoit les documents, celui-ci enregistre dans une base de données (EAN-Document Mapping Table) les correspondances entre un document et des numéros EAN. Cette table permettra par la suite de savoir quels documents sont disponibles pour telles personnes.

L'European Article Number[10] (EAN) est un code-barres de 13 chiffres qui permet dans notre cas d'identifier les médecins.

2.5.5 Partie client

Au niveau du client, une application tierce nommée MediWay est employée par le médecin privé. Le but de la partie cliente du prototype est de faire le pont entre l'OHT Bridge et MediWay afin de récupérer des documents. Le code source de MediWay n'étant pas disponible, la solution mise en place fut la création d'une librairie externe qui se charge d'envoyer les requêtes au Bridge.

Pour ce qui est de la partie hôpital, les fichiers sont transmis au prototype via FTP comme cité dans la section précédente. Il serait envisageable d'installer la petite application Java directement sur un serveur de l'hôpital.

2.5.6 Processus de soumission d'un document

Afin de bien comprendre le processus de soumission d'un document par l'hôpital, un schéma est disponible dans l'annexe 15.2. Celui-ci montre en détail les différents composants impliqués et quels sont leurs rôles.

2.5.7 Processus de récupération d'un document

Dans l'annexe 15.3, un schéma détaille le processus de récupération d'un document par le médecin privé.

2.6 Documents émis par l'organe Cybersanté Suisse

La Suisse travaille sur sa stratégie Cybersanté depuis 2006. Sa stratégie nationale n'a débuté qu'assez tard, en partie dû à son système de santé fragmenté. Chaque canton a son propre système de santé et règlement. En conséquence, la stratégie Cybersanté ne peut que formuler des recommandations aux cantons.

L'organe national de coordination en matière de cybersanté Confédération – cantons (organe de coordination « Cybersanté Suisse ») est opérationnel depuis début 2008. Le Conseil fédéral a adopté en janvier 2006 un nouveau chapitre intitulé « Santé et système de la santé » lors de la révision de la « Stratégie pour une société d'information en Suisse ». Puis en septembre 2007, la Confédération et les cantons ont signé une convention cadre de collaboration de la « Stratégie Cybersanté Suisse ». Ils ont ainsi montré leur volonté de s'unir et ont créé l'organe de coordination Confédération – cantons [18].

[10] http://www.gs1.org/

Le projet partiel « Normes et architecture[11] » de l'organe de coordination élabore des recommandations relatives à l'architecture cybersanté [19]. L'organe de coordination a fixé des principes de base, puis en a déduit les principaux éléments de l'architecture :

- Réseau sûr pour l'échange de données
- Index des patients
- Index des fournisseurs de prestation
- Registre des documents
- Archivage des documents
- Système d'autorisation
- Portail d'accès
- Points de transmission du système

Ces recommandations sont illustrées dans l'image 6.

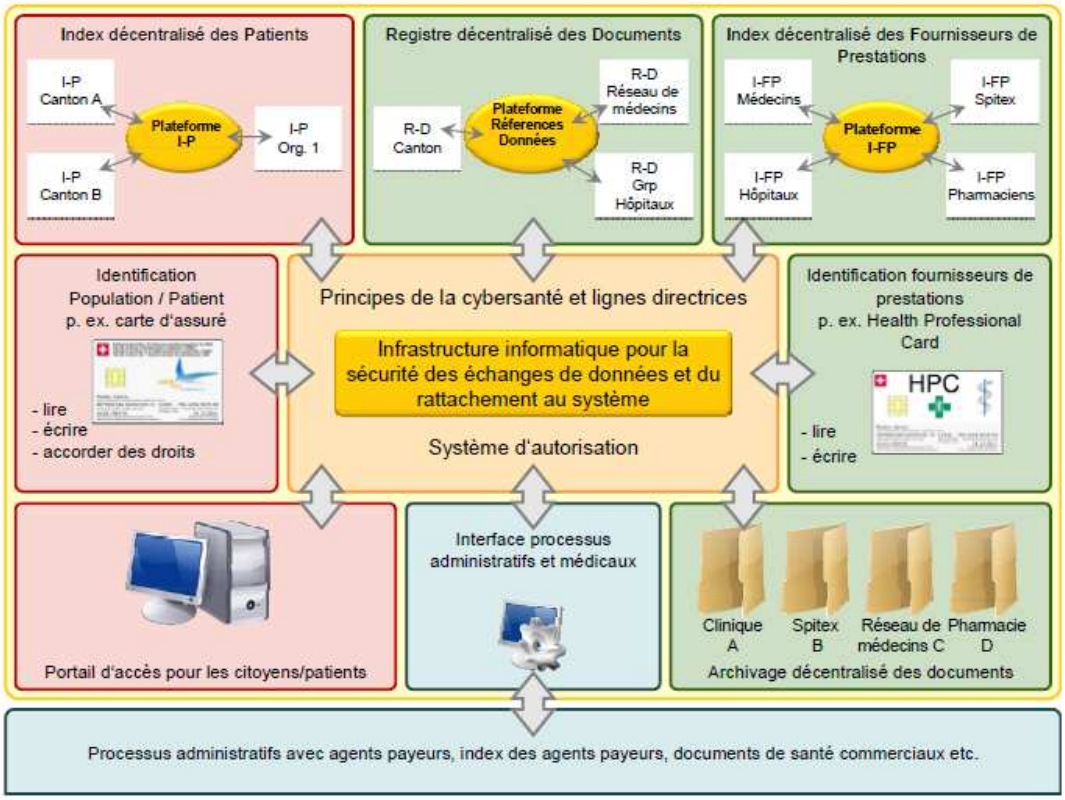


Image 6 : Recommandations eHealth Suisse

La stratégie eHealth Suisse a fixé des buts jusqu'en 2015 et espère voir la réalisation de plateformes basées sur des standards médicaux tels qu'HL7 et IHE.

[11] http://www.e-health-suisse.ch/umsetzung/00146/00148/index.html?lang=fr&download=NHzLpZig7t,lnp6I0NTU042l2Z6ln1ae2IZn4Z2qZpnO2Yuq2Z6gpJCDdHt4g2ym162dpYbUzd,Gpd6emK2Oz9aGodetmqaN19XI2IdvoaCVZ,s-

3 Audit de sécurité

Dans cette seconde phase, les aspects de sécurité du prototype seront examinés. Une première approche fut la réunion du 27.05.2010 au RSV de Sion qui introduisait le projet MediCoordination aux participants, puis une présentation technique a été réalisée. Certains points relatifs à la sécurité ont été discutés pour bien comprendre le mécanisme du prototype et ainsi donner quelques suggestions d'amélioration. Par la suite, un rapport de l'analyse de sécurité du prototype a été transmis par le responsable de la sécurité du RSV, dans lequel il relève les incompréhensions et incohérences de différents documents et propose des améliorations possibles. Enfin, le travail accompli dans le domaine de la sécurité sera mis en évidence et les points faibles seront soulevés.

3.1 Réunion du 27.05.2010 au RSV Sion

Cette réunion a eu lieu dans le cadre du projet MediCoordination où travaille en collaboration des personnes de la HES-SO et le Service informatique du RSV de Sion. Lors de cette réunion, les personnes suivantes étaient présentes : Alexandre Gnaegi, Michel Buri, Franck Boisde, Bruno Alves, Michael Schumacher et moi-même.

La réunion a débuté avec une introduction au projet MediCoordination par M. Schumacher, puis par une présentation technique du prototype par M. Alves. Dans les travaux à venir, M. Alves propose d'implémenter le « Secure node » du profil ATNA. Ceci permettrait l'authentification des différents acteurs du prototype, ainsi qu'une meilleure gestion des droits d'accès. De plus, le profil ATNA se charge de loguer tous les évènements du système, tels que les login, logout et accès à des fichiers.

Par la suite, une discussion sur certains aspects de sécurité du prototype a commencé. Le principal point récurrent est l'authentification du client (médecin généraliste). Actuellement, seul le serveur a un certificat, pas le client. Toutes les requêtes passent par un proxy qui les filtre par rapport aux adresses IP autorisées, donc seul les clients avec une adresse IP autorisée peuvent lancer des requêtes sur le serveur.

Pour conclure, des améliorations possibles ont été proposées et sont listées ci-dessous :

- Séparer l'OHT Bridge des services Registry / Repository
- Crypter les fichiers des crédentiels sur le serveur
- Offusquer le connecteur client (DLL intégrée à MediWay)
- Réduire le nombre de technologies utilisées (SOAP 1.1 – 1.2)
- Transmettre des documents sur le répertoire de manière cryptée (FTP)
- Authentifier de manière sûre le client

Pour l'authentification du client, une proposition est de mettre en place des certificats pour une authentification mutuelle du client - serveur. Une seconde proposition est l'utilisation de la carte FMH[12] qui permet de s'authentifier, ainsi que de signer et crypter un document [20].

3.2 Rapport d'analyse de sécurité

Quelques jours après la réunion, le responsable de la sécurité informatique du RSV, M. Buri, nous a transmis un rapport d'analyse de sécurité du prototype MediCoordination. Ce rapport se base sur plusieurs sources d'information que sont :

- Le rapport « Deliverable 4 : MediCoordination Prototype[13] »
- La présentation « MediCoordination – Pratical approach to interoperability between hospitals and general practitioners in Switzerland » de M. Schumacher
- La présentation « MediCoordination – Technical presentation of the prototype application » de M. Alves
- La séance de présentation du projet MediCoordination lors de la journée HIL du 8 mars 2010 à l'ICHV / Sion
- La séance de présentation du projet MediCoordination au Service informatique du RSV du 27 mai 2010 à l'ICHV / Sion

De plus, certains documents disponibles sur le site du projet MediCoordination ont également été consultés.

Cette analyse se basant uniquement sur une revue documentaire et aucunement sur preuve physique, technique ou encore confirmative, est de nature purement déclarative. Malgré l'impossibilité de donner un avis fondé sur la sûreté du prototype, une analyse a été réalisée.

[12] http://www.fmh.ch/fr/services/cps_hpc.html

[13] http://www.medicoordination.ch/index.php?option=com_content&view=article&id=51&Itemid=62

3.2.1 Critères d'analyse

Pour évaluer le prototype, plusieurs critères légaux et institutionnels ont été fixés. Pour la base juridique, la loi fédérale sur la protection des données (LPD[14]) et l'ordonnance relative à la loi fédérale sur la protection des données (OLPD[15]) ont été retenues. Les grands principes considérés dans le cadre de cette analyse sont :

- La proportionnalité : La collecte et le traitement de données doit s'effectuer seulement en cas de nécessité pour un but ou une fonction spécifique.
- La finalité : Les données personnelles ne doivent être traitées que dans le but qui est indiqué lors de leur collecte.
- Le devoir d'informer en cas de collecte de données sensibles : Ce principe précise que lors d'une collecte de données, les personnes concernées doivent être informées au minimum du maître du fichier, des finalités du traitement et des catégories de destinataires lors de l'enregistrement ou de la première communication à un tiers, sauf si la collecte est prévue par une loi.
- La sécurité : La LPD instaure l'obligation de sécuriser les données personnelles. Plus une donnée est sensible, plus la sécurité doit être importante. Au niveau de l'OLPD, les contours du système de sécurité sont définis, en prévoyant des normes générales protégeant les données des menaces sur leur disponibilité, intégrité, authenticité et confidentialité.

Ces quatre points représentent les critères légaux d'analyse.

Des critères techniques d'analyse ont également été fixés. Pour ce faire, les recommandations émise dans le projet-cadre « Normes et architecture[16] » d'eHealth Suisse ont été prises en compte. Les éléments suivants ont été considérés dans cette analyse :

- Réseau sûr pour l'échange de données
- Index des patients
- Index des fournisseurs de prestations
- Système d'autorisation
- Système de traçabilité médico-légale
- Point de transmission du système

3.2.2 Résultat de l'analyse

Malgré la difficulté d'effectuer une analyse basée uniquement sur une revue documentaire, ainsi que les incohérences et imprécisions de certains documents rendant relativement difficile une compréhension suffisante du contexte de réalisation du projet, une première analyse est donnée ci-dessous.

[14] http://www.admin.ch/ch/f/rs/c235_1.html
[15] http://www.admin.ch/ch/f/rs/c235_11.html
[16] http://www.e-health-suisse.ch/hinweise/index.html?lang=fr

3.2.2.1 Les critères légaux d'analyse

- Proportionnalité : Les fichiers envoyés sont des lettres de sortie. Ces documents contiennent peu de données, mais sont malgré tout considérés comme sensibles.
- Finalité : Il peut être raisonnablement admis que la finalité de la transmission des données vers les médecins privés a été définie par l'équipe de projet.
- Devoir d'informer en cas de collecte de données sensibles : Le RSV n'informe pas de manière explicite les patients dont les données sont transmises aux médecins privés dans le cadre du projet MediCoordination. Il est cependant admissible d'étendre le consentement obtenu d'un patient lors de l'admission à l'hôpital à une transmission électronique de l'avis de sortie à son médecin traitant.
- Sécurité : Les points relatifs à la sécurité sont traités ci-dessous.

3.2.2.2 Les critères techniques d'analyse

- Réseau sûr pour l'échange de données
    - Le cryptage des documents PDF n'est effectué qu'à l'arrivée sur le serveur par le Bridge et non lors de l'envoi. Ces documents transitent en clair sur canal sécurisé par du TLS (Transport Layer Security), ce qui représente une faille de sécurité.
    - Les connexions entre le client et le serveur sont cryptées par TLS. Cependant, seul le serveur détient un certificat, alors que le client n'est pas clairement authentifié. De plus, les connexions entre le Bridge et le Registry / Repository ne sont pas sécurisées.
    - Le prototype utilise le protocole WS-Security pour l'authentification de la personne. Les messages SOAP sont cryptés grâce à TSL. Toutefois, il n'est pas précisé pourquoi le protocole WS-Security n'est pas employé à la place de TLS.
    - Evaluer la possibilité de segmenter la zone démilitarisée du RSV et de créer une zone spécifique. Cette remarque est destinée au Service informatique du RSV.
- Index des patients
    - Actuellement, le prototype ne gère pas d'index des patients. La seule notion de patient est un id placé dans les fichiers CDA. Le profil PIX n'est pas implémenté pour le moment.
- Index des fournisseurs de prestations
    - Aucune information relative aux fournisseurs de prestation n'est fournie par une base de données dans le cadre du projet MediCoordination.

- Système d'autorisation
    - Le protocole WS-Security est utilisé pour l'authentification des utilisateurs du système. Les crédentiels (username, password, EAN) sont intégrés aux en-têtes des messages qui sont ensuite vérifiés par l'OHT Bridge. Il conviendrait de préciser si les mots de passe sont hachés et de décrire les appels utilisant ce protocole.
    - Le fichier contenant les relations « username/password/EAN » est stocké en clair sur le serveur, ce qui représente une faille de sécurité importante.
- Système de traçabilité médico-légale
    - La traçabilité médico-légale n'est pas implémentée dans le prototype. Les documents transmis par l'hôpital ne sont pas digitalement signés.
- Point de transmission du système
    - Pour l'intégration du prototype au logiciel MediWay, une librairie a été développée et mise en GAC (Global Assembly Cache) et constitue donc une faille de sécurité. Des recommandations existent sur la sécurisation de librairie en GAC.
    - La politique de purge des répertoires de travail sur l'ordinateur du médecin-privé n'est pas gérée par le prototype.
    - Il serait recommandé de transférer les documents de manière sécurisée, malgré le fait que l'accès est restreint à l'intranet du RSV.
    - Il serait conseillé de séparer les documents des métadonnées dans plusieurs bases de données.
    - Il faudrait analyser les avantages du mode « Stored Query » par rapport à celui « Query Registry transaction ».
    - Les services IIP sont hébergés sur un poste de travail. L'installation sur une architecture appropriée est recommandée.

En plus de ces différentes remarques, plusieurs suggestions ont été données, telles que l'implémentation de certains profils IHE et la rédaction d'un chapitre dédié aux aspects sécurité.

3.3 Analyse personnelle de la sécurité du prototype

Dans cette partie, une analyse personnelle des aspects de sécurité du prototype est effectuée. Dans un premier temps, la sécurité globale du prototype est analysée. Ensuite, certains aspects sont revus en détail et enfin une comparaison des différents mécanismes pour sécuriser des messages SOAP est réalisée.

3.3.1 Vue globale de la sécurité du prototype

Tous les aspects de sécurité du prototype sont repris dans cette section. L'Image 7 présente les communications entre les composants du serveur, ainsi que les différents protocoles et outils utilisés.

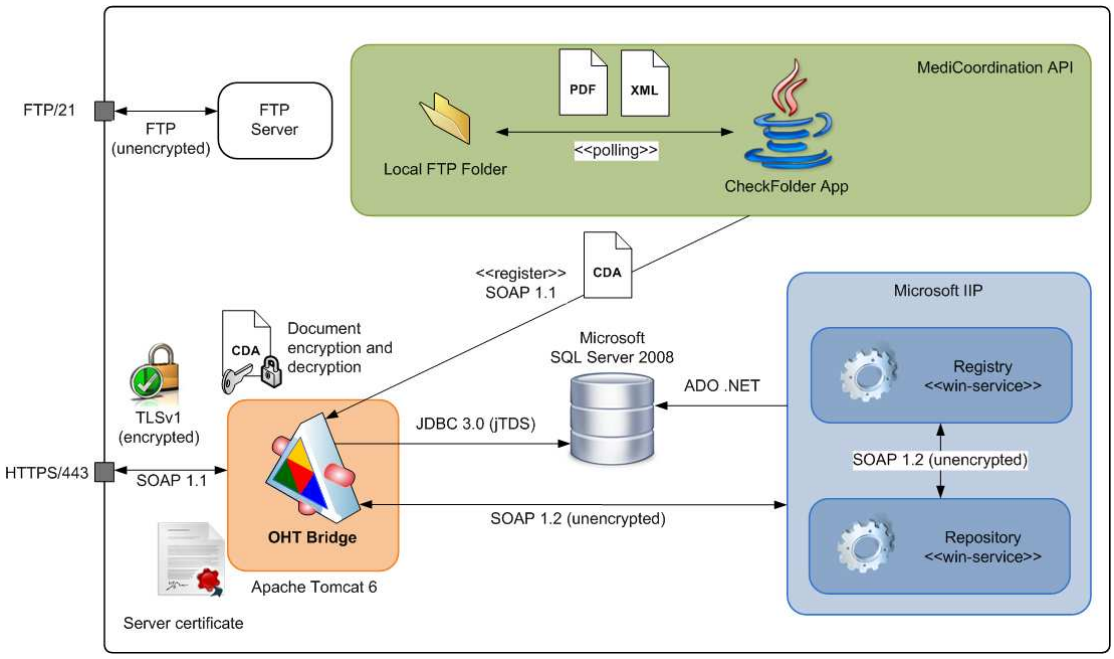


Image 7: Vue globale de la sécurité du prototype

Sur le serveur du prototype, les ports 443 et 21 sont ouverts. Le port 21 correspond au serveur FTP sur lequel sont déposés les fichiers PDF et XML. La connexion vers ce serveur n'est pas sécurisée, mais est accessible uniquement depuis le domaine de l'hôpital. Le port 443, quant à lui, correspond au web service accessible depuis Internet. Les communications sont cryptées par le protocole TLS v1 de manière unilatérale, seul le serveur a un certificat signé par une entité de certification personnelle. Les algorithmes de cryptage supportés sont « TLS_RSA_WITH_AES_128_CBC_SHA » et « TLS_RSA_WITH_NULL_SHA ». Les autres connexions, notamment entre le bridge, les services IIP et la base de données MSSQL ne sont pas sécurisées. Dans le cas du prototype, ces communications sont locales et ne présentent donc pas une grande faille de sécurité. Cependant, dans le cas d'une séparation de ces différents composants sur des serveurs différents, ces communications doivent être sécurisées [21].

La plupart des éléments relatifs à la sécurité ont été relevés dans l'audit du responsable de la sécurité informatique du RSV et ne sont donc pas repris ici. Néanmoins, la gestion des authentifications et des autorisations des utilisateurs est reprise en détail dans la section 3.3.2, ainsi que la sécurisation des messages SOAP dans la section 3.3.3.

3.3.2 Gestion des authentifications et des autorisations

L'authentification des utilisateurs s'effectue par le contrôle des crédentiels avec ceux de la base de données. Les informations stockées sont le login, le hash SHA-1 du mot de passe, ainsi que le numéro EAN. Les crédentiels sont intégrés aux en-têtes des messages SOAP grâce au profil WS-UsernameToken et sont vérifiés par un mécanisme de Tomcat[17]. C'est donc le Bridge qui s'occupe de l'authentification des utilisateurs.

Il est important de noter que cette vérification s'effectue lors de communications entre un client et le web service. En observant de plus près l'Image 7, il faut constater que les requêtes du médecin-privé passent par le port 443 et ainsi chaque message SOAP est vérifié. En ce qui concerne la soumission d'un document par l'hôpital, c'est l'application Java qui s'occupe de transmettre les documents. De ce fait, il est impossible de savoir quelles personnes ont déposé quels fichiers.

Le système d'autorisation du prototype correspond au numéro EAN associé à un document, ainsi une personne ne peut récupérer que les documents enregistrés avec son numéro EAN. Cette façon de faire apporte peu de flexibilité au système. La mise en place d'un système d'autorisation basé sur les rôles (RBAC – Role-Based Access Control) permettrait de prendre une décision d'accès à une ressource basée sur le rôle de l'utilisateur. Cette gestion aide ainsi à regrouper plusieurs utilisateurs sous le même rôle et facilite donc la maintenance pour l'administrateur [22].

Une solution pour résoudre le problème de l'authentification et des autorisations serait l'utilisation de SAML et XACML. Ces deux standards sont les prédécesseurs de WSS et SAML est repris par celui-ci pour résoudre ces problèmes dans le cadre d'architecture SOA (Service Oriented Architecture).

SAML (Security Assertion Markup Language) est un standard basé sur XML pour l'échange de données en lien avec l'authentification et l'autorisation entre deux domaines de sécurité. SAML a été produit par OASIS. Le but de SAML est d'apporter l'authentification unique (Single Sign-On) entre plusieurs domaines. SAML permet l'expression d'assertion concernant la confiance sous forme de tokens XML dans l'en-tête de messages SOAP. Ainsi, le service recevant ces messages peut prendre une décision quant aux droits d'accès à une ressource. Cependant, il ne remplace pas l'authentification dans la mesure où celle-ci est nécessaire avant d'émettre un token SAML qui peut contenir le sujet de l'authentification et même des autorisations [23].

[17] http://tomcat.apache.org/

XACML (eXtensible Access Control Markup Language) est un langage basé sur XML qui définit le contrôle d'accès, la circulation des règles et de l'administration de la politique de sécurité d'un système d'information [24]. L'architecture d'XACML est présentée dans l'Image 8.

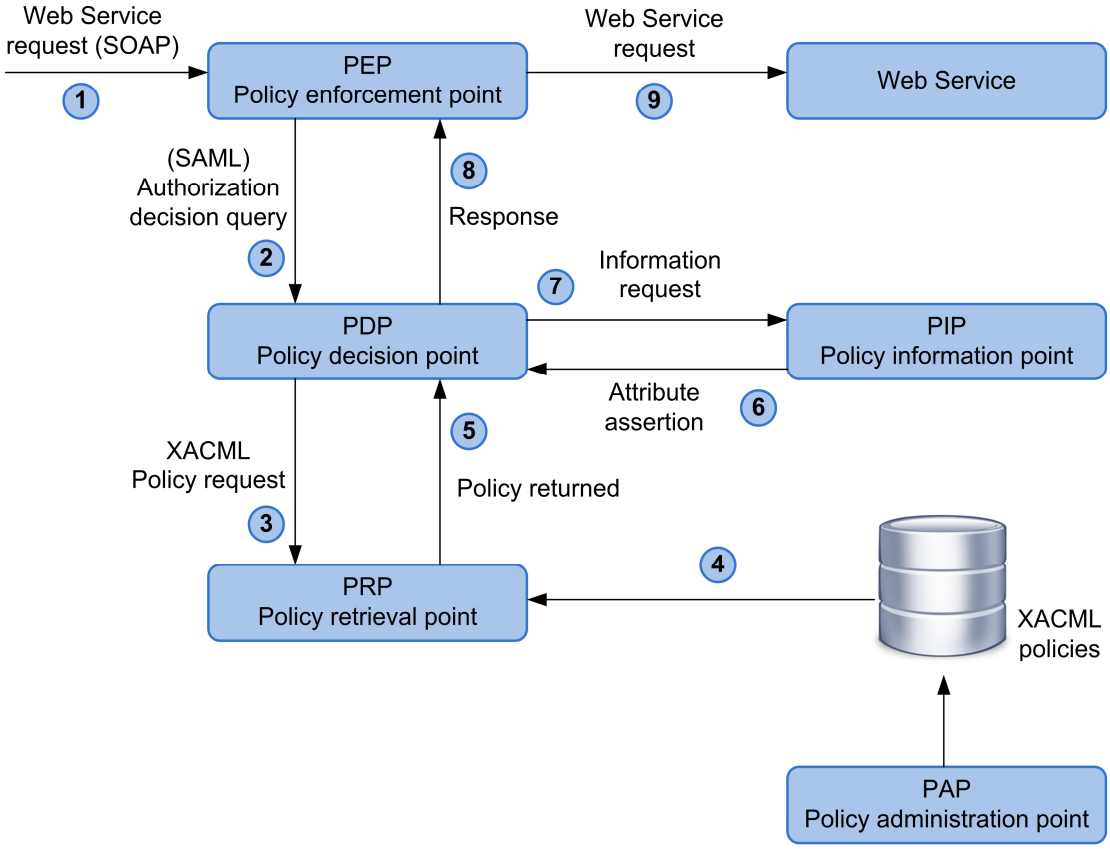


Image 8 : Architecture d'XACML

Les différents composants de cette architecture sont :

- PEP (Policy enforcement point)
- PDP (Policiy decision point)
- PRP (Policy retrieval point)
- PAP (Policy administration point)
- PIP (Policy information point)

Lors de l'arrivée d'une demande vers le web service, le PEP l'intercepte et crée une requête afin de prendre une décision d'autorisation (1). La réponse à cette requête détermine si la demande est autorisée ou non. Cette requête contient les informations relatives à l'émetteur de la demande et est envoyée vers le PDP (2). Ce dernier, comme son nom l'indique, décide si la demande est autorisée ou refusée. Afin de pouvoir prendre une décision, le PDP doit récupérer les règles XACML. Celles-ci sont soit internes (PDP), soit externes et nécessitent donc l'utilisation du PRP (3-5). Le PAP est le point d'entrée pour administrer les règles de sécurité. Après avoir

récupéré les règles d'autorisation, PDP peut vérifier le prédicat d'une règle par rapport à un attribut, par exemple au rôle de l'utilisateur dans un système RBAC. L'attribut peut être récupéré en interrogeant le PIP, par exemple un serveur Kerberos (6-7). Enfin, le web service reçoit la demande avec l'ajout de la décision d'autorisation (9) [25].

Avec ce système, la gestion des authentifications et des autorisations est plus souple et permet de mieux satisfaire les recommandations de l'environnement métier. L'utilisation de SAML autorise l'envoi d'autres informations en rapport à la sécurité de l'utilisateur final et protège contre les attaques par rejeu (replay attack). De plus, XACML permet l'autorisation basée sur les rôles. Ainsi, plusieurs docteurs auraient accès à une ressource sans devoir le préciser pour chacun.

3.3.3 Différences entre WS-Security et TLS

Lors de la conception de l'architecture d'un système, la question « Quels moyens de sécurité mon système a-t-il besoin ? » devrait être posée. Dans le cas du prototype MediCoordination, le point central repose sur un web service. Plusieurs solutions sont disponibles pour la sécurisation des messages SOAP.

SOAP (Simple Object Acces Protocol) est un protocole de RPC (Remote Procedure Call) orienté objet bâti sur XML. SOAP permet la transmission de messages entre objets distants. Ce transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également s'effectuer par un autre protocole, comme SMTP. SOAP est notamment utilisé dans les web services [26].

Actuellement, les communications entre le client et le web services via des messages SOAP sont sécurisées par le protocole TLS. Transport Layer Security (TLS) est un protocole de sécurisation des échanges sur Internet. Ce protocole s'appelait anciennement Secure Sockets Layer (SSL) [27].

Les web services de première-génération utilisent principalement HTTP, celui-ci pouvant être sécurisé par TLS. Dans le cas d'une seule connexion entre un client SOAP et un web service, TLS est le choix évident pour apporter la confidentialité et l'authentification. Pour quelles raisons le protocole TLS ne suffit-il pas pour sécuriser ces communications ? La première raison est que le protocole SOAP est indépendant du protocole de communication qu'il utilise. Plusieurs technologies de communication peuvent être employées dans le cadre de messages SOAP, telles que HTTP, SMTP, HTTPR et Jabber. TLS ne permet qu'une sécurité de point à point, alors que si les messages SOAP traversent une multitude de web services, une sécurité dite « end-to-end » est nécessaire. De plus, si un web service souhaite réaliser des actions en rapport avec les accréditations de l'utilisateur final, celui-ci doit avoir accès aux

informations d'authentification / autorisation de l'utilisateur. TLS se situant sur la couche 5 du modèle OSI, il ne fournit pas ces informations [28].

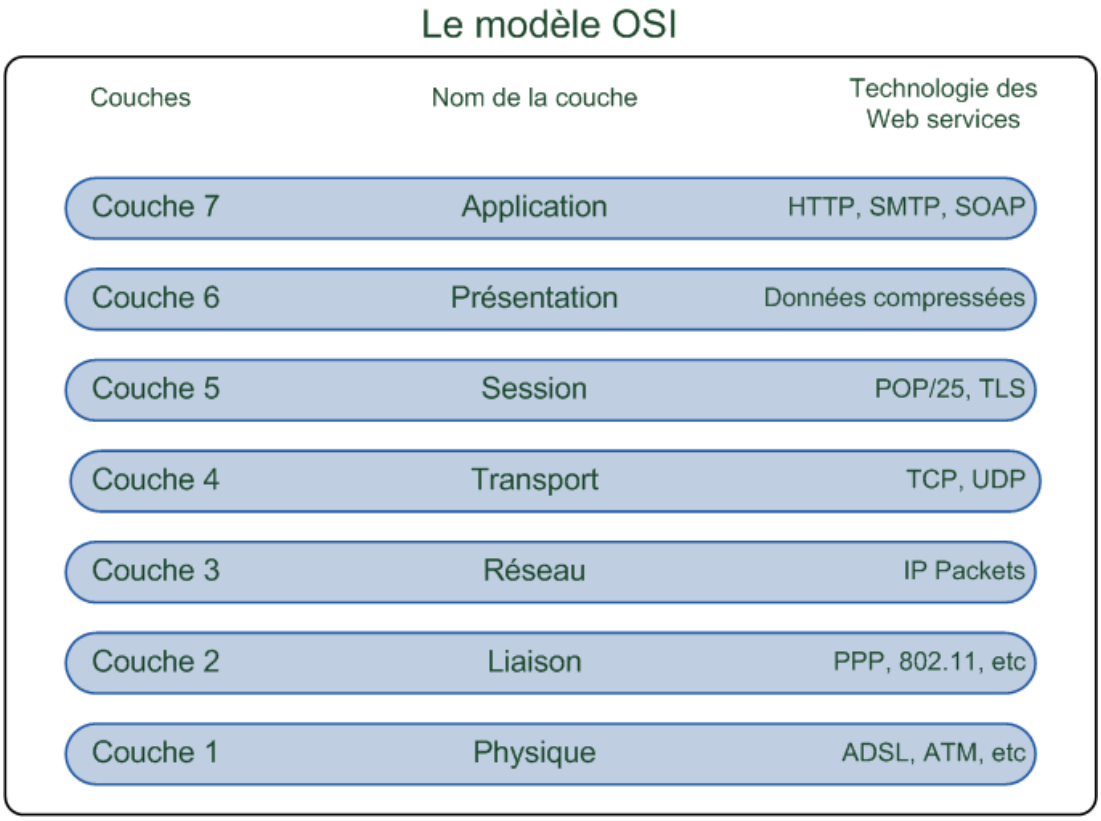


Image 9 : Le modèle OSI

Le modèle OSI (Open Systems Interconnection) est un modèle de communications entre ordinateurs proposé par l'ISO (Organisation internationale de normalisation). Il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions [29].

Afin de résoudre ces problèmes, WS-Security a été développé. WS-Security (WSS) est un framework qui permet d'appliquer de la sécurité aux web services. Ce dernier permet d'inclure des informations de sécurité au format XML dans un message SOAP. WSS explique comment utiliser des standards existants, tels que XML Signature et XML Encryption, afin d'appliquer la confidentialité et l'intégrité aux messages des web services [30].

3.3.3.1 XML Signature

XML Signature est une spécification produite conjointement par le W3C (World Wide Web Consortium) et l'IETF (Internet Engineering Task Force). XML Signature permet de signer digitalement des portions d'un document XML, mais également de n'importe quel format de données.

PKCS#7 est un moyen pour crypter et signer des données et est le prédécesseur de l'XML Signature et XML Encryption. PKCS#7 utilise l'ASN.1 (Abstract Syntax

Notation number 1) connu pour sa complexité. De plus, un interpréteur ASN.1 est nécessaire pour produire et vérifier des signatures PKCS#7.

Une importante caractéristique de l'XML Signature dans les web services est sa capacité à pouvoir signer uniquement des données sélectionnées. Ainsi, un seul paramètre d'un message SOAP peut être signé et assure alors l'intégrité « end-to-end » en cas de transport par plusieurs intermédiaires [31].

3.3.3.2 XML Encryption

XML Encryption est une spécification du W3C. Il apporte un moyen de crypter des parties d'un document XML, mais également n'importe quelles données, puis rend les données cryptées au format XML.

XML Encryption n'est pas le remplacement de TLS. Ce dernier est toujours le choix de base pour la confidentialité entre deux entités qui communiquent via HTTP. Cependant, si ce contexte s'étend à plusieurs machines n'utilisant pas forcément HTTP, l'XML Encryption est idéal pour la confidentialité [32].

3.3.3.3 WS-SecureConversation

Après avoir vu l'XML Signature et l'XML Encryption qui sont la base de WSS, le profil WS-SecureConversation est évalué. Comme dit précédemment, TLS est largement utilisé pour l'authentification point-à-point et pour la confidentialité des web services (sécurité au niveau transport). Pour le trafic SOAP qui traverse plusieurs intermédiaires, une sécurité niveau message est exigée. C'est pourquoi, WS-Security définit comment utiliser des tokens de sécurité en collaboration avec l'XML Signature et l'XML Encryption. Ces tokens sont évalués à chaque message SOAP reçu, ce qui pose un problème de performance dû au manque de la notion de session pour un groupe de messages.

Pour pallier à ce problème, WS-SecureConversation permet à un client et un web service de s'authentifier mutuellement via des messages SOAP et d'établir un contexte de sécurité. Comme TLS, WS-SecureConversation construit le contexte de sécurité avec des clés asymétriques afin de négocier une clé symétrique. Ainsi, chaque message SOAP ne devra pas être vérifié.

WS-SecureConversation est employé pour échanger de manière sûre un contexte de sécurité. Ce profil est conçu pour les messages SOAP qui traversent plusieurs intermédiaires. L'utilisation de WS-SecureConversation n'exclut pas une sécurité niveau transport pour les liens point-à-point [33].

3.3.3.4 Conclusion

Après avoir observé les différences entre TLS et WS-Security et plus particulièrement WS-SecureConversation, l'utilisation de WSS semble plus appropriée. WSS sécurise au niveau message alors que TLS le fait au niveau transport. WSS permet de signer et crypter que des parties spécifiques du message SOAP et permet le routage entre plusieurs intermédiaires. La seule zone d'ombre dans l'utilisation de WSS est la charge produite par l'XML Signature et l'XML Encryption lors d'échange fréquent de messages. Le profil WS-SecureConversation peut réduire cette charge et fournit une sécurité « end-to-end ». Le Tableau 1 indique le nombre de messages par seconde pouvant être envoyé pour les différents mécanismes de sécurité [34].

Security Mechanism	**Messages/second**
WS-Security (X.509) XML Signature & Encryption	352
WS-SecureConversation XML Signature & Encryption	798
Transport Layer Security (TLS)	2918

Tableau 1 : Performance pour la sécurité des messages SOAP

TLS apporte la confidentialité et l'authentification entre un client SOAP et un web service. Ce scénario correspond à l'implémentation du prototype, ce qui peut expliquer l'utilisation de TLS. Si l'utilisation d'informations de sécurité sur l'utilisateur final n'est pas nécessaire, ainsi que l'implémentation du routage SOAP dans le futur n'est pas requis, TLS est une solution pragmatique. Cependant, l'utilisation du profil WS-UsernameToken démontre le besoin d'accéder aux informations liées à l'utilisateur final. Un système sûr doit être protégé par WSS et par TLS. Il faut protéger le message tout en assurant l'intégrité et la confidentialité du transport. Le passage vers les nouvelles technologies pour la sécurité des web services sera certainement nécessaire dans le futur [35].

4 Analyse des profils IHE

Dans cette troisième partie, une analyse des profils IHE est effectuée pour résoudre les failles de sécurité relevées dans la partie « Audit de sécurité ». Plusieurs profils IHE répondent à des besoins métiers ou d'implémentation. Des améliorations seront proposées à la fin de cette analyse, telles que des choix techniques, l'ajout de profils IHE, ainsi que des recommandations.

Après recherche des aspects de sécurité dans les profils IHE, il s'avère que les mécanismes de sécurité ne sont pas complétement définis par les profils IHE d'infrastructure. IHE suppose que certaines conditions d'implémentation telles que la sécurité de l'environnement physique (équipement situé dans une zone protégée, impossibilité de connecter un ordinateur directement sur le réseau) et la sécurité de l'environnement réseau (mise en place d'un firewall, connections sécurisées) sont en place. Les attaques réseau, d'infection par un virus ne sont pas prises en charge par les profils IHE.

Le profil d'intégration XDS ne prescrit pas de politiques de sécurité pour une bonne raison. Ces politiques de sécurité dépendent du type de système de santé, ainsi que les lois du lieu de l'implémentation, ce qui offre une plus grande flexibilité à XDS [36].

4.1 Solutions

Après avoir reçu le rapport d'analyse de sécurité, plusieurs documents ont été modifiés pour combler les lacunes relevées dans l'audit de sécurité. Des incohérences ont été corrigées et l'ajout d'une section « Security » dans le fichier D4 apporte un plus pour la compréhension globale de prototype.

Les remarques de la réunion du 27.05.2010 ont été prises en considération. Les failles relevées lors de cette réunion sont reprises dans le rapport d'analyse de sécurité et ne sont donc pas directement citées.

Trois points n'ont pas été pris en compte lors cette analyse car ils ne concernaient pas directement le prototype MediCoordination, mais sont de la responsabilité du RSV :

- La segmentation de la zone démilitarisée
- Le transfert FTP de manière sécurisée
- L'hébergement des services IIP sur une architecture appropriée

Tous les points de l'audit de sécurité sont repris ici et des propositions sont amenées afin d'améliorer la sécurité du prototype. Certains profils IHE sont cités dans les solutions, ceux-ci seront repris en détail à la section 4.3.

4.1.1 Les critères légaux d'analyse

4.1.1.1 Proportionnalité

Les documents transmis dans le cadre du projet MediCoordination consistent en un fichier PDF et des métadonnées. Le PDF est un avis de sortie envoyé par l'hôpital vers le médecin privé. Un document contient :

- La date du traitement
- Des informations générales
  - Titre du document
  - Temps du séjour à l'hôpital
  - Code de confidentialité
- Des informations liées au patient
  - Identifiant du patient
  - Nom, prénom, date de naissance et adresse
- Des informations concernant l'expéditeur
  - Identifiant du professionnel de la santé
  - Information personnelle
  - Organisation représentée
- Des informations sur les destinataires
  - Les mêmes informations que l'expéditeur en ajoutant les EAN correspondants

Ces informations sont considérées comme sensibles. Cependant, le principe de proportionnalité est respecté de mon point de vue. Uniquement les données relatives à la lettre de sortie sont transmises et non pas le dossier complet du patient.

4.1.1.2 Finalité

L'analyse a admis que la finalité de la transmission des données a été définie par l'équipe de projet.

4.1.1.3 Devoir d'informer en cas de collecte de données sensibles

Le consentement obtenu d'un patient lors de l'admission à l'hôpital peut être raisonnablement étendu à une transmission électronique de l'avis de sortie. Si dans le futur un consentement écrit du patient est exigé, le profil BPPC (Basic Patient Privacy Consents) répondrait à ce besoin.

4.1.2 Les critères techniques d'analyse

4.1.2.1 Réseau sûr pour l'échange de données

Les aspects de sécurité évoqués dans le cadre de ce critère sont liés aux données qui transitent entre le client et le serveur. Pour résoudre le problème des documents PDF cryptés, un mécanisme pour crypter et décrypter doit être mis en place au niveau du client. Du côté du médecin-privé, le connecteur .Net pourrait effectuer ce travail,

mais du côté de l'hôpital, les fichiers sont déposés dans un dossier directement via FTP, le développement d'un composant qui crypterait le PDF et l'enverrait, serait donc requis. Le canal de communication étant actuellement sécurisé avec le protocole TLS, il est justifiable dans le cas d'un prototype d'investir les ressources disponibles dans d'autres éléments.

Le second aspect relevé est le cryptage des communications entre les différents acteurs du prototype. Actuellement, aucune sécurité n'est mise en place pour les messages entre le Bridge et le Registry / Repository. Une authentification mutuelle avait été définie au début du projet, puis abandonnée faute de temps. Il faut quand même noter que ces deux composants se situent sur la même machine physique, ce qui réduit le risque. Pour ce qui est des connexions entre le client et le serveur, le protocole TLS est utilisé. Cependant, seul le serveur détient un certificat, le client n'étant donc pas complètement authentifié. Un proxy est actuellement en place pour filtrer les requêtes vers les serveurs grâce aux adresses IP du client. Il serait envisageable d'intégrer un système d'authentification mutuelle par certificat ou l'implémentation du profil ATNA qui permettrait d'authentifier chaque nœud du système.

La sécurisation des messages SOAP s'effectue via TLS. Dans le protocole WS-Security, le profil WS-UsernameToken est employé pour transmettre les crédentiels avec chaque requête. L'ajout du profil WS-SecureConversation permettrait d'encrypter les corps des messages SOAP. Pour plus de détail, il faut consulter la section 3.3.3.

4.1.2.2 Index des patients

Comme dit précédemment, aucun index de patients n'est géré par le prototype. L'établissement de listes décentralisées de patients pour une identification sans équivoque des personnes en traitement dans différents systèmes de santé est une priorité. Dans le projet « Normes et architecture », ehealth Suisse propose l'utilisation de la carte d'assuré en tant que moyen d'identification et d'authentification pour les patients. Cette carte permet également le stockage de données. En ce qui concerne IHE, le profil PIX (Patient Identifier Cross-referencing) permettrait de répondre à ce problème.

4.1.2.4 Index des fournisseurs de prestations

Aucune base de données concernant les fournisseurs de prestations n'est employée dans le cadre du projet MediCoordination. L'utilisation de la carte HPC (Health Professional Card) est recommandée par eHealth Suisse. Elle permet l'identification et l'authentification des fournisseurs de prestations, ainsi que l'encryptage et la signature de document. L'implémentation du profil XUA (Cross Enterprise User Assertion) ferait le lien entre les utilisateurs des différents systèmes afin d'avoir une sorte d'index des fournisseurs de prestations.

4.1.2.5 Système d'autorisation

L'authentification des utilisateurs du système s'effectue via le protocole WS-Security, et plus particulièrement sur le profil WS-UsernameToken. Les crédentiels (username, hash du password, EAN) sont placés dans l'en-tête des messages SOAP. Lors de la réception des messages SOAP, le Bridge vérifie la conformité des crédentiels.

Les relations « username/password/EAN » sont contenues dans un fichier stocké en clair sur le serveur. Après la réception du rapport d'analyse de sécurité, ce fichier a été remplacé par une table dans une base de données.

Le rapport suggère d'évaluer une implémentation du profil EUA (Enterprise User Authentication).

La gestion des autorisations a été détaillée dans la section 3.3.2 et l'utilisation de SAML et XACML avait été proposée comme solution. Le projet e-Toile de Genève emploie ces deux standards, notamment grâce à OpenSSO[18]. OpenSSO est une plateforme open source de management des accès. Il fut créé par Sun Microsystems et suite au rachat de ce dernier par Oracle, il est développé par ForgeRock sous le nom d'OpenAM. Une liste de différents produits implémentant SAML est consultable sur le site saml.xml.org[19].

4.1.2.6 Système de traçabilité médico-légale

Actuellement, seuls les logs du Bridge et du serveur XDS peuvent être utilisés pour détecter des erreurs. Une suggestion serait d'implémenter le profil ATNA qui met en place plusieurs mécanismes de sécurité, dont la création de logs.

4.1.2.7 Point de transmission du système

L'ajout d'une librairie en GAC constitue une faille de sécurité. Après la réunion du 27.05.2010, cette librairie a été digitalement signée pour améliorer le niveau de sécurité. Une amélioration possible serait d'offusquer la librairie afin de ralentir la rétro-conception.

[18] https://opensso.dev.java.net/

[19] http://saml.xml.org/wiki/saml-open-source-implementations

En ce qui concerne la politique de purge des répertoires de travail sur l'ordinateur du médecin-privé, après que les documents ont été téléchargés, le prototype n'a aucun contrôle sur le cycle de vie de ces fichiers. Il conviendrait d'en discuter avec les développeurs de MediWay.

Les métadonnées et les documents devraient être stockés sur des serveurs de bases de données différents. Dans une architecture réelle, le service Registry se situerait sur un serveur différent du service Repository, ainsi les bases de données seraient séparées. Dans le cas d'un prototype, la simplicité est de mise pour gagner du temps et ne pas compliquer l'architecture.

Le dernier point de ce critère consiste à analyser les avantages du mode «°Stored Query°» par rapport au «°Query Registry transaction°». Ces deux modes correspondent aux transactions possibles entre l'acteur Document Consumer et le Repository. Le mode «°Query Registry transaction°» fait référence au profil XDS.a et n'est pas supporté par XDS.b. Le prototype se basant sur XDS.b, c'est le mode « Stored Query » qui est utilisé. Stored Query, signifiant procédure stockée en français, est un mécanisme pour laisser la logique métier sous forme de commandes côté serveur. Ce système permet d'éviter l'exécution de codes malicieux sur le serveur.

4.2 Recommandations

Un chapitre dédié à la sécurité a été ajouté au document D4 après réception du rapport d'analyse de sécurité. Dans ce chapitre, tous les aspects de la sécurité du prototype sont abordés, tels que les algorithmes de cryptage des communications, les ports ouverts du serveur, ainsi que le système d'authentification. Cela permet une meilleure compréhension du travail accompli dans ce domaine.

Afin d'améliorer encore l'image du prototype en matière de sécurité, quelques suggestions sont amenées ici. Une description de la méthode de travail et des mesures prises pour améliorer la sécurité permettra de valoriser le travail accompli dans ce domaine.

Une configuration particulière d'un protocole afin d'améliorer la sécurité devrait être relevée. Le système d'authentification doit être clairement défini et les restrictions en matière de mot de passe citées. Il faudrait préciser la présence de règles de codage spécifiques aux aspects de sécurité. Notre prototype se basant principalement sur un web service, il faut tenir compte des recommandations de projets tels qu'OWASP[20] (Open Web Application Security Project).

[20] http://www.owasp.org/index.php/Main_Page

OWASP est une organisation à but non-lucratif dont l'objectif est d'améliorer la sécurité des applications web. Chaque année, OWASP publie un document nommé « OWASP Top 10[21] » qui présente les 10 failles les plus courantes et leurs contre-mesures.

De plus, des tests d'intrusion peuvent être effectués pour observer si des failles sont détectées. Des distributions open-source comme Backtrack[22] offre une panoplie d'outils pour tester la robustesse d'un système et plus spécifiquement, des frameworks comme Metasploit[23] permettent de lancer des attaques de manière automatique pour découvrir des failles de sécurité. Enfin, la mise en place d'un système de logs donnerait la possibilité d'un audit ultérieur afin d'observer des activités inhabituelles.

4.3 Spécification des profils IHE

4.3.1 Profil BPPC

Le profil Basic Patient Privacy Consents fournit un mécanisme pour l'enregistrement du consentement du patient. Ainsi, des documents publiés sur un serveur XDS peuvent être marqués par ce consentement qui prouve l'autorisation du patient quant à la publication d'un document. De plus, ce profil force l'acteur XDS Consumer à traiter les documents avec respect de la vie privée du patient. Le profil XDS donne une petite marche à suivre pour supporter la confidentialité des patients à l'intérieur de l'XDS Affinity Domain. C'est par exemple le cas dans le prototype MediCoordination avec l'utilisation d'un champ « confidentiallityCode » dans les fichiers CDA échangés. Par contre, aucune information n'est fournie pour savoir comment utiliser ce champ dans le cadre de la confidentialité d'un patient. Le profil BPPC complète XDS en décrivant un mécanisme pour implémenter de multiples politiques de confidentialité à l'intérieur d'un XDS Affinity Domain, ainsi que l'intégration des contrôles d'accès pour les acteurs XDS [37].

4.3.2 Profil PIX

Le profil Patient Identifier Cross-referencing permet de référencer les identifiants des patients de plusieurs domaines. Par la suite, un système pourra corréler les informations d'un patient avec les données d'autres systèmes connaissant le patient par un identifiant différent. Ceci permet d'avoir une vue plus complète des informations d'un patient.

L'intégration du profil PIX dans une implémentation d'XDS est observable dans Image 10 : Intégration du profil PIX.

[21] http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202010.pdf
[22] http://www.backtrack-linux.org/
[23] http://www.metasploit.com/

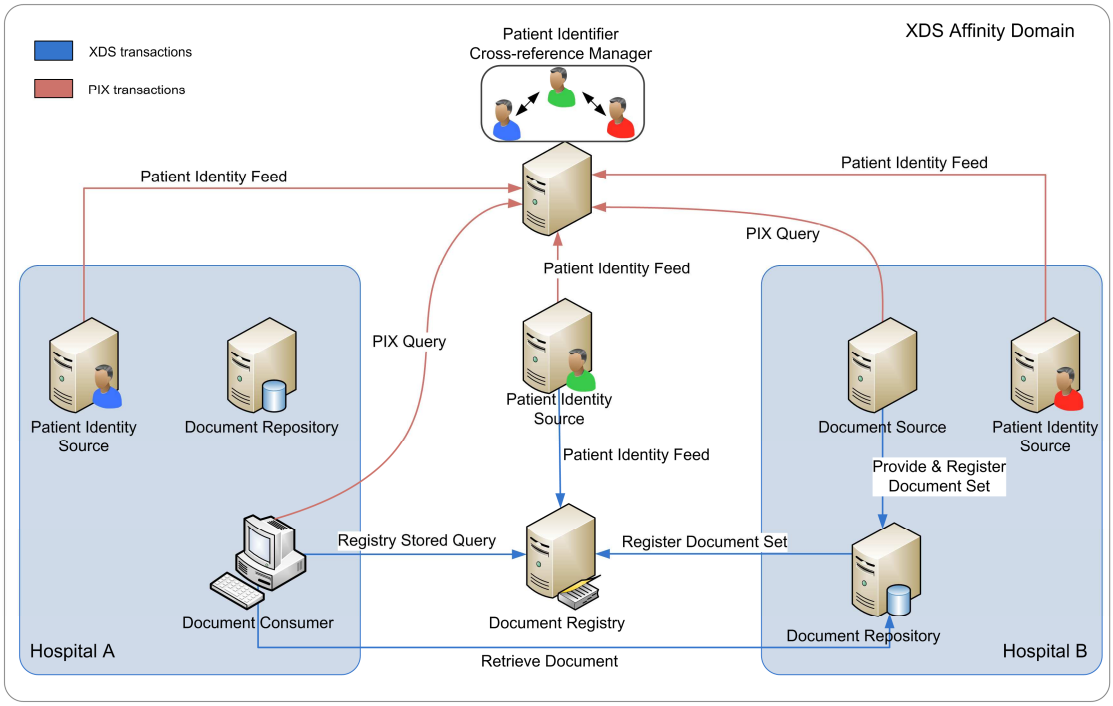


Image 10 : Intégration du profil PIX

Trois acteurs « Patient Identity Source » sont observables ici. Le premier correspond aux identifiants des patients utilisés par un hôpital A, le second aux identifiants des patients au niveau du XDS Affinity Domain et le dernier, aux identifiants de l'hôpital B. Chaque acteur « Patient Identity Source » envoie des notifications pour chaque événement en lien avec les identifiants du système à l'acteur « Patient Identifier Cross-reference Manager ». Cet acteur est chargé de garder un lien entre les identifiants de chaque sous-système correspondant à un même patient. Par la suite, l'acteur « Patient Identifier Cross-reference Consumer » l'interroge afin de récupérer des documents sur d'autres systèmes [38].

4.3.3 Profil XUA

Le profil Cross-Enterprise User Assertion fournit un moyen de communiquer l'identité d'une personne lors de transactions inter-entreprises. Afin de gérer ces transactions, il y a un besoin d'identifier la personne requérante pour permettre au système receveur de prendre des décisions pour les contrôles d'accès et également de générer des logs cohérents.

Des profils IHE pour l'authentification d'un utilisateur tel que le profil EUA, ne sont pas destinés pour fonctionner dans un cadre inter-entreprises. Les transactions entre deux entreprises maintenant leur propre système d'authentification est géré par le profil XUA. Celui-ci utilise le protocole Web-Services Security et SAML 2.0 Token Profile pour l'identification inter-entreprises [39].

L'intégration du profil XUA dans une implémentation XDS est représentée dans Image 11 : Intégration du profil XUA.

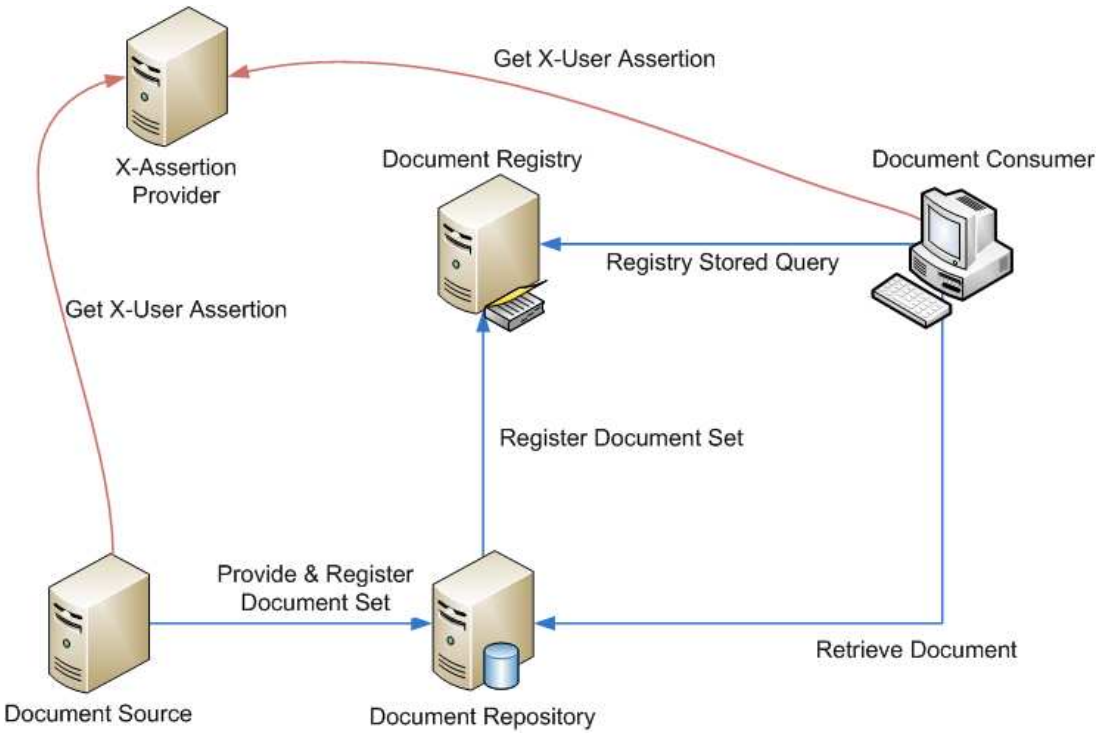


Image 11 : Intégration du profil XUA

Dans ce cas d'implémentation, les acteurs Document Source et Document Consumer intègrent également les caractéristiques de l'acteur X-Service User définies dans le profil XUA. L'acteur X-Service provider est quant à lui implémenté par le Document Registry et Document Repository. Lors d'une requête pour enregistrer ou récupérer un document, l'acteur X-Service User demande une sorte de confirmation à l'acteur X-Assertion Provider. Dès lors, cet acteur peut interroger les acteurs X-Service provider. Dans le cas des logs pour le profil ATNA, les acteurs génèrent des logs contenant l'identifiant local ainsi que l'ajout du profil XUA [40].

4.3.4 Profil EUA

Ce profil définit un moyen d'établir un seul nom par utilisateur qui par la suite est utilisé par les périphériques et logiciels du système. Ce profil recommande l'utilisation du protocole Kerberos, ainsi que le standard HL7 CCOW (Clinical Context Object Workgroup) pour la partie « user subject ». Cela incite à un management centralisé des utilisateurs et apporte la commodité d'un Single Sign-On. Ce profil se base principalement sur les caractéristiques du protocole Kerberos, telles que l'authentification par un système de question/réponse pour éviter de transmettre des mots de passe sur le réseau, l'obtention d'un ticket d'accès à un service et les communications sécurisées. Le standard HL7 CCOW fournit la notion de

« User Context » qui permet à une application d'utiliser les crédentiels de l'utilisateur sans que celui-ci doive les réentrer. Ce profil s'applique pour la gestion des utilisateurs d'une seule entreprise et ne doit pas s'étendre à travers internet.

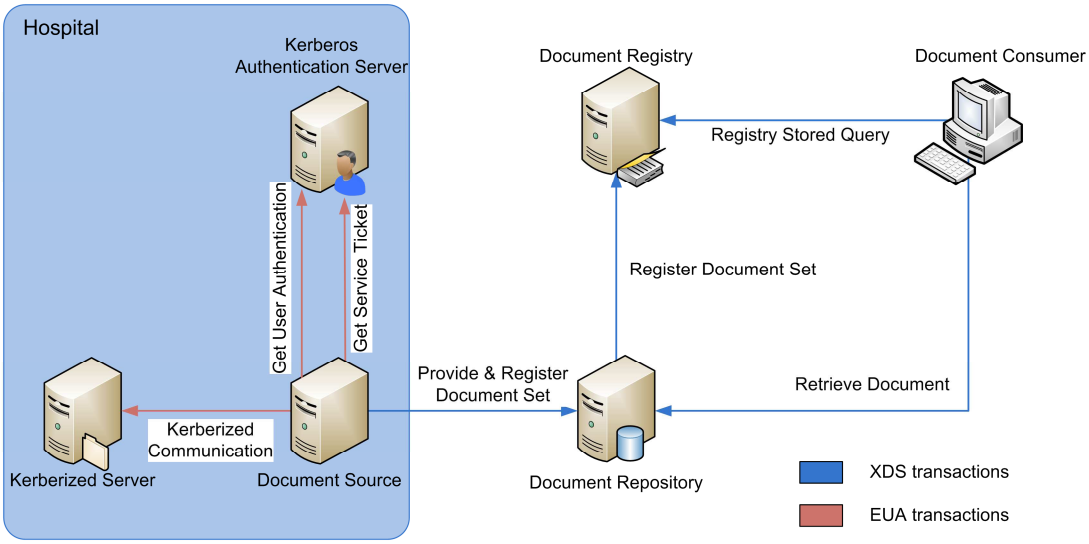


Image 12 : Intégration du profil EUA

L'implémentation du profil EUA dans un hôpital est représentée par l'Image 12. Un serveur Kerberos est utilisé pour l'authentification des utilisateurs. Après le processus d'authentification, un utilisateur peut demander un ticket pour accéder à des serveurs, par exemple un serveur de fichiers. De plus, par le biais de l'User Context, les crédentiels peuvent être employés pour d'autres transactions IHE.

4.3.5 Profil CT

Le profil Consistent Time définit un mécanisme pour la synchronisation du temps des systèmes d'un même domaine. Plusieurs systèmes de sécurité et profils IHE requièrent l'utilisation d'une base commune pour le temps. Ce profil spécifie l'utilisation du protocole NTP (Network Time Protocol) défini par la RFC1305. Ce système permet une synchronisation ayant moins d'une seconde d'erreur. Ceci est nécessaire par exemple lors de la création de logs afin d'avoir une cohérence dans les événements et également pour le protocole Kerberos qui exige moins de 5 minutes de différence entre le client et le serveur.

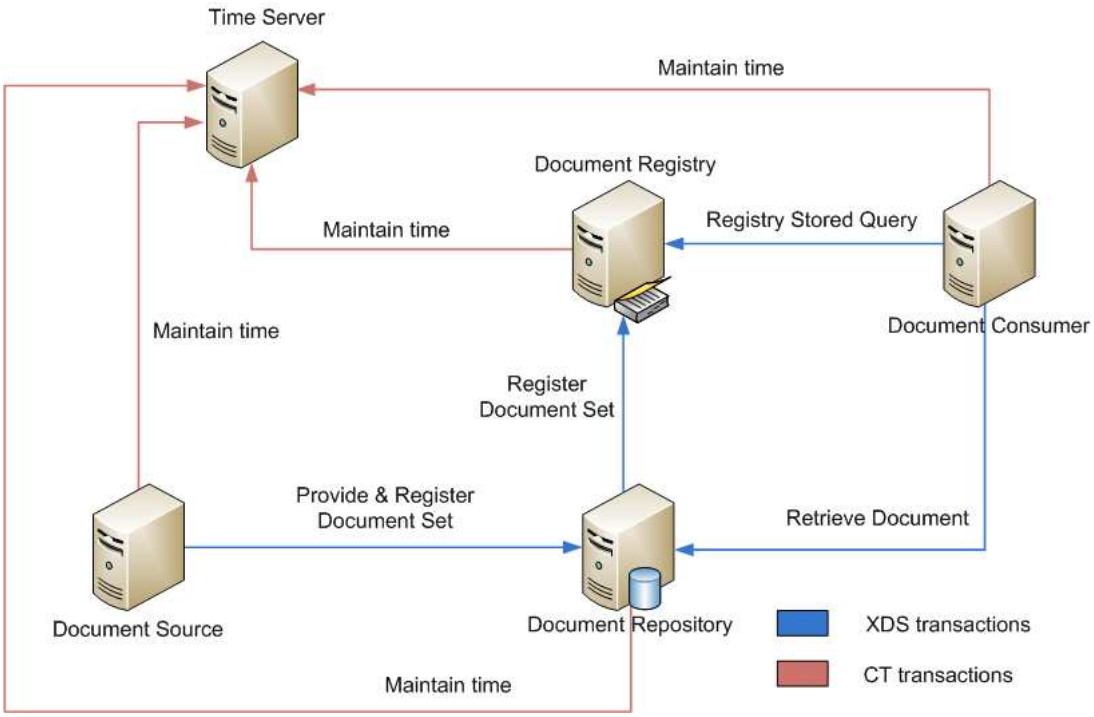


Image 13 : Intégration du profil CT

Lors de l'ajout du profil CT dans une implémentation du profil XDS (Image 13), chaque machine du système est synchronisée avec le serveur NTP afin de maintenir son horloge interne en fonction de l'heure de ce dernier.

4.3.6 Profil ATNA

Le profil Audit Trail and Node Authentication établit des mesures de sécurité qui fournissent la confidentialité des informations, l'intégrité des données et les contrôles d'accès aux utilisateurs. Cet environnement est considéré comme le domaine de sécurité (Domain Security) et peut s'étendre par exemple au domaine d'affinité XDS. Le profil ATNA exige que les points suivants soient respectés :

- Toutes les machines sont authentifiées. Cette authentification identifie la machine comme faisant partie du domaine de l'entreprise.
- L'identification de la machine est utilisée pour déterminer quels sont les accès qui devraient être autorisés sur cette machine, ainsi que les accès de la personne utilisant cette machine.
- Chaque « secure node » est responsable de fournir un contrôle d'accès. Ceci inclut l'authentification et les autorisations des utilisateurs.
- Chaque « secure node » est responsable de générer des logs de sécurité. Dans le domaine de la santé, les logs sont souvent plus utiles qu'un accès strict aux données.

Ce modèle a été créé dans l'hypothèse d'un échange de documents entre plusieurs machines. Malgré les implications de sécurité de ce modèle, le profil ATNA assume que certains aspects de sécurité soient en place, tels que la protection contre les attaques réseau, les infections par un virus et la formation du personnel [41].

Ce profil est divisé en deux grandes parties : l'authentification et les logs.

4.3.6.1 Authentification

Le profil ATNA contribue au contrôle d'accès en limitant l'accès réseau entre les nœuds et en limitant l'accès de chaque nœud aux utilisateurs autorisés. Pour l'authentification des utilisateurs, chaque nœud peut employer la technologie de son choix pour le contrôle d'accès. Le profil EUA est un exemple, mais d'autres choix sont possibles, tels que la biométrie ou un simple username/password.

En ce qui concerne l'authentification des nœuds, le profil exige l'utilisation de certificats pour une authentification bidirectionnelle entre chaque nœud. Les standards DICOM et HL7, ainsi que le protocole HTML (HyperText Markup Language) ont des mécanismes définis d'authentification basés sur les certificats. DICOM et HL7 ne sont pas des protocoles, mais fournissent des méthodes pour des communications sécurisées. Dans le cas de DICOM, celui-ci précise l'utilisation IPSec ou TLS, les deux utilisant des certificats. Le profil ATNA recommande l'utilisation du protocole TLS pour les connexions entre les nœuds n'étant pas sur un réseau sûr.

Les certificats X.509 sont recommandés par IHE. L'organisation locale (XDS Affinity Domain) doit déterminer quel mécanisme sera utilisé pour l'authentification. Deux choix sont possibles :

- Validation des certificats basée sur la signature d'un CA (Certificate Authority). Le principe est que le certificat est signé par un tiers de confiance. Ce procédé est appelé « Chain of trust ».
- Validation des certificats via des certificats autorisés. Sur chaque nœud sont intégrés les certificats qui seront par la suite acceptés pour l'authentification.

Dans le cas d'un certificat non valable, le système devrait interdire ou restreindre les communications avec certains nœuds.

Le prototype MediCoordination utilise un web service pour interroger les services Registry/Repository. Ce web service est considéré comme le point d'entrée au système. Le profil ATNA précise l'utilisation du profil WS-I Basic Security Version 1.1[24] pour l'établissement des associations de sécurité [42].

[24] http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html

4.3.6.2 Logs

Les logs fournissent la traçabilité des événements du système. Ces logs sont stockés dans une base de données nommée « Audit Record Repository ». Par la suite, cette base de données est interrogée par un responsable de sécurité pour détecter si des événements ne sont pas conformes à la politique de sécurité du domaine. Par le biais de ces informations, il devrait être possible de pouvoir répondre aux questions suivantes :

- Pour un utilisateur, quels sont les documents interrogés ?
- Pour un document, quels sont les utilisateurs qui l'ont interrogé ?
- Quelles mauvaises authentifications des utilisateurs ont été reportées ?
- Quelles mauvaises authentifications des machines ont été reportées ?

Deux formats sont retenus pour la représentation d'un message de log. Il s'agit de la RFC3881 (Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications) et du standard DICOM Supplement 95. La RFC3881 définit un schéma XML pour reporter des événements de sécurité. DICOM Supplement 95 étend le vocabulaire de base proposé par la RFC3881 et y ajoute certains champs optionnels.

Le transport des logs est défini par l'utilisation du « Reliable Syslog Cooked Profile » (RFC3195) ou le BSD Syslog (RFC3164). Le BSD Syslog a plusieurs limitations, d'où la recommandation d'utiliser la RFC 3195.

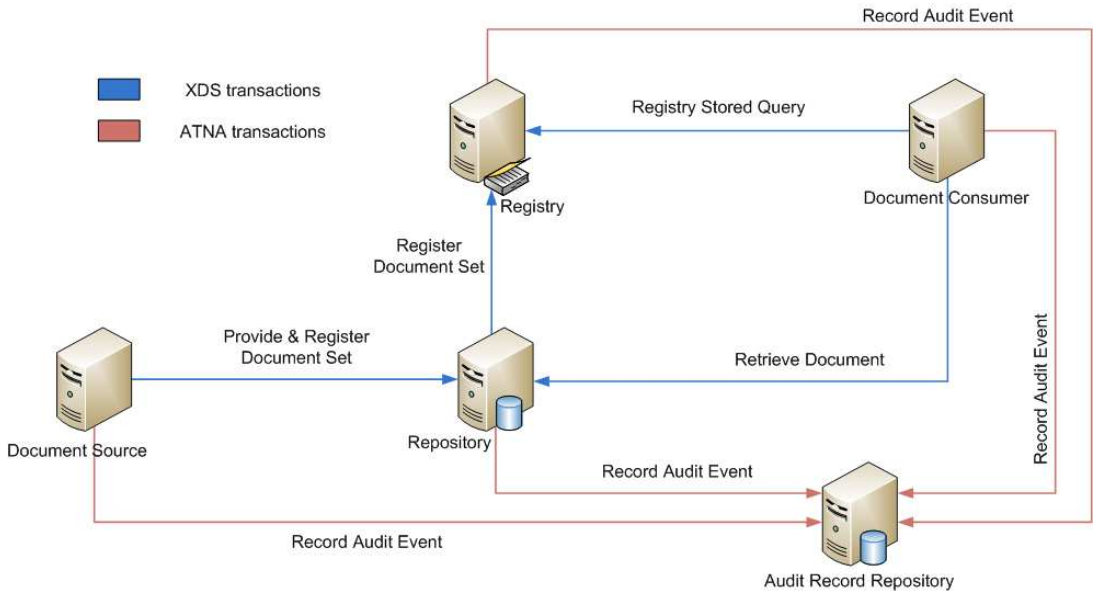


Image 14 : Intégration du profil ATNA

Dans l'Image 14, chaque nœud envoie des logs à la base de données (Audit Record Repository). Dans ce schéma, les éléments liés au profil CT ont été volontairement omis pour une meilleure lisibilité.

4.4 Priorité d'implémentation

Dans cette section, les modifications à apporter au prototype sont triées par priorité. Les critères utilisés pour l'évaluation sont les suivants :

- le risque encouru : Plus une faille de sécurité est critique, plus il devient urgent de la corriger.
- la valeur ajoutée : Dans le cas d'implémentation de profil IHE, la valeur ajoutée doit être prise en compte.
- la proportionnalité : Ce critère prend en compte les différences entre prototype et implémentation réelle. Certaines remarques sont justifiées, mais ne sont pas prioritaires dans le cadre d'un prototype.
- la faisabilité : L'ajout ou la modification d'un composant doit être réalisable de manière raisonnable.

La liste ci-dessous désigne les implémentations à réaliser dans le cadre du prototype MediCoordination :

1. Implémentation des profils ATNA et CT : Le profil ATNA résoudra les problèmes que sont l'authentification des nœuds, la sécurité des communications et la traçabilité des événements.
2. Mise en place d'une meilleure gestion des authentifications et autorisations : L'authentification des utilisateurs s'effectue via le profil WS-Username Token, puis les autorisations sur un document via l'EAN. Un système plus flexible (SAML - XACML) permettrait une meilleure adaptation aux besoins du domaine médical.
3. Ajout du profil WS-SecureConversation pour sécuriser les communications au niveau message.
4. Intégration des profils PIX et XUA : L'implémentation de ces deux profils est un passage obligatoire vers une architecture plus conséquente. Cependant, l'utilisation des cartes d'assurés et HPC requiert un appareil pour leur lecture. L'équipe MediCoordination ne dispose pas de ces lecteurs de cartes actuellement.
5. Offusquer la libraire .Net : Un premier pas a été réalisé en signant numériquement la librairie. En l'offusquant, la rétro-conception serait plus difficile et inciterait un pirate à trouver un système plus vulnérable.
6. Séparation des bases de données et des différents services : Dans le cadre du prototype, tous les services et bases de données se situent sur le même serveur pour un déploiement plus simple et pour limiter les coûts. Dans l'intention d'implémenter ce système dans une architecture concrète, les services sont de toute manière séparés.

8. Crypter le PDF du côté client : La remarque de crypter les documents du coté client est pertinente. Cependant, la transmission des documents est sécurisée par le protocole TLS ce qui dans le cadre d'un prototype est suffisant de mon point de vue.
9. Politique de purge des répertoires : Le prototype n'a aucun accès sur la durée de vie des documents transmis. Ce point devrait être éclairci avec les développeurs de MediWay.
10. Mise en place du mode Stored Query : Ce mode est déjà en place au niveau des services Registry/Repository. De plus, les clients n'ont pas directement accès aux services, mais passent leurs requêtes via le web service.

Il faut prendre en considération le cadre de l'implémentation du prototype. Dans le cas d'échange de documents, le profil XDS détermine la notion d'XDS Affinity Domain. Celui-ci est un groupe d'entreprises qui se sont mises d'accord sur les politiques de sécurité, le format d'échange des documents et les règles métier. Dans l'implémentation du prototype, l'acteur Document Source (hôpital) et Document Consumer (médecin-privé) sont déjà en place et l'équipe MédiCoordination n'y a pas accès. Il est alors difficile de modifier ces parties du prototype.

Cependant, il faut garder à l'esprit que l'implémentation concrète sera bien différente de celle du prototype. Le Registry sera partagé par plusieurs entreprises et plusieurs repository seront disponibles. Les bases de données seront distribuées sur différents sites. Cela implique des communications via internet d'où le besoin de sécuriser les communications et l'authentification des nœuds. Une implémentation concrète est démontrée dans l'Image 15.

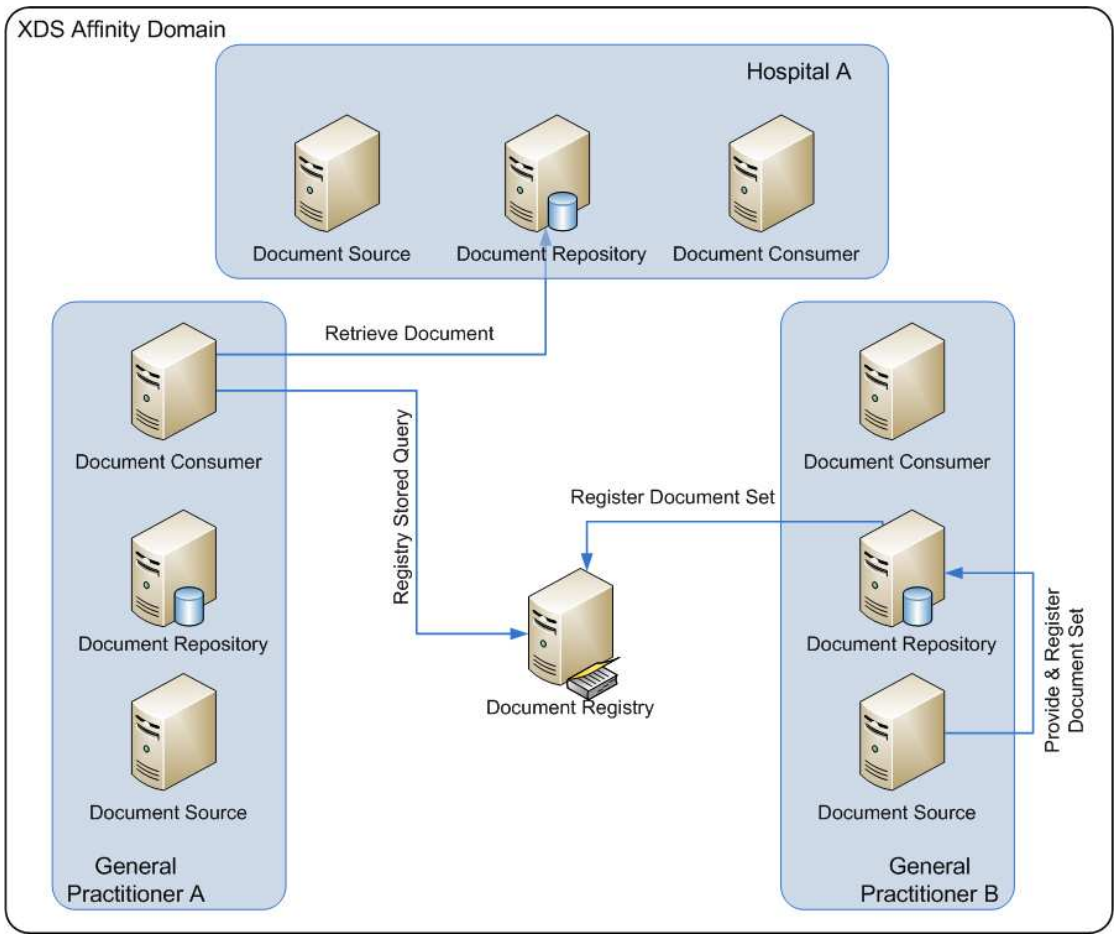


Image 15 : Implémentation concrète du prototype

4.5 Choix

L'implémentation des profils ATNA et CT a été retenue pour la suite du travail. Celle-ci renforcera la sécurité de prototype et permettra un audit ultérieur pour l'ajustement des règles de sécurité.

Afin d'intégrer le profil ATNA et CT au prototype MediCoordination, une recherche d'outils existants a été effectuée. Iheprofiles d'OHT implémente les transactions du profil ATNA. Après vérification, il semblerait que les services IIP de Microsoft gèrent également les transactions du profil. Le seul composant qui manque est l'acteur Audit Record Repository. Mes recherches se sont donc orientées vers cet acteur. Sur le site d'IHE[25], il est possible d'observer les résultats des Connectathons IHE.

Le Connectathon IHE est un événement à grande échelle pour l'industrie IT de la santé afin de tester l'interopérabilité de leurs produits. Les Connectathons ont lieu annuellement en Asie, Europe et Amérique du Nord. Cette manifestation dure une semaine durant laquelle des systèmes de vendeurs différents exécutent des

[25] http://sumo.irisa.fr/con_result/

transactions relatives aux acteurs IHE sélectionnés. Par la suite, les résultats sont publiés pour valider l'interopérabilité et la conformité d'un produit avec les profils IHE [43].

Les résultats démontrent que plusieurs grandes entreprises IT, telles que Siemens, Oracle et IBM, se sont impliquées dans le domaine de l'eSanté. Cependant, ces entreprises vendent leurs services et le projet MediCoordination se base actuellement sur des outils open source uniquement. Par la suite, deux projets ont été trouvés. Le premier se nomme metu-ihe-atna project[26] et fournit une implémentation open source du profil ATNA. Cependant, celui-ci semble abandonné depuis quelques temps déjà. Le second s'appelle OpenATNA[27] développé par l'Université de Cardiff et les solutions Open Sources de Misys (MOSS). OpenATNA est une implémentation de l'acteur Audit Record Repository. Celui-ci supporte le format de messages de la RFC3881 via le transport BSD Syslog. OpenATNA fait partie des projets supportés par OpenHealthTools. Il sera donc employé pour la partie serveur du profil ATNA et ainsi tous les composants du profil seront définis.

En ce qui concerne le profil CT, un serveur NTP doit être mis en place dans l'architecture du prototype. Les systèmes d'exploitation Windows et Linux permettent de monter un serveur NTP relativement facilement.

[26] http://sourceforge.net/projects/metu-ihe-atna/files/

[27] https://openatna.projects.openhealthtools.org/

5 Design des profils sélectionnés

Dans cette partie, l'architecture du prototype est modifiée afin d'y intégrer les profils ATNA et CT sélectionnés dans la section « Analyse des profils IHE ». Dans un premier temps, une approche théorique est effectuée afin d'associer le profil CT et ATNA avec le profil XDS. Ensuite, ce schéma est reporté sur le prototype dans le but de déterminer les modifications à apporter.

5.1 Schéma d'intégration des profils ATNA et CT

Avant de s'intéresser au prototype, une vue théorique de l'insertion des profils ATNA et CT au profil XDS est apportée par l'Image 16.

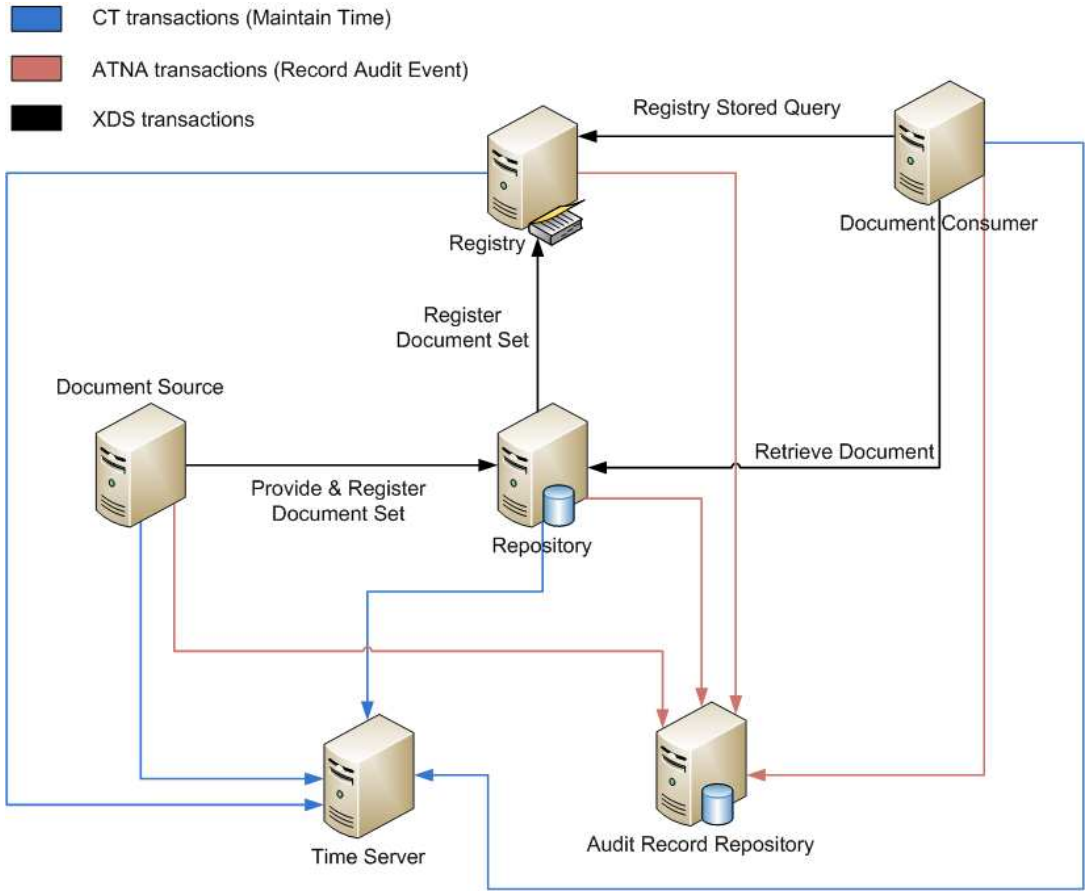


Image 16 : Schéma théorique des profils ATNA, CT et XDS

Chaque acteur du profil XDS gère les transactions « Maintain Time » et « Record Audit Event ». Chaque machine synchronise son horloge interne via le serveur NTP. Le besoin d'avoir un temps synchronisé se fait sentir lors de la génération de logs afin d'y avoir une cohérence chronologique. Ces logs sont envoyés vers l'acteur Audit Record Repository qui sera par la suite interrogé par un responsable de sécurité.

Un aspect pas encore abordé est l'authentification des nœuds via des certificats. Chaque acteur « Secure Node » doit, en plus de maintenir le temps et de générer des logs, pouvoir s'authentifier de manière mutuelle avec un autre nœud. Pour se faire, des certificats doivent être générés et placés sur les acteurs Secure Node. L'Image 17 met en évidence les machines Secure Node et les authentifications mutuelles. Les communications entre les Secure Nodes sont sécurisées via le protocole TLS.

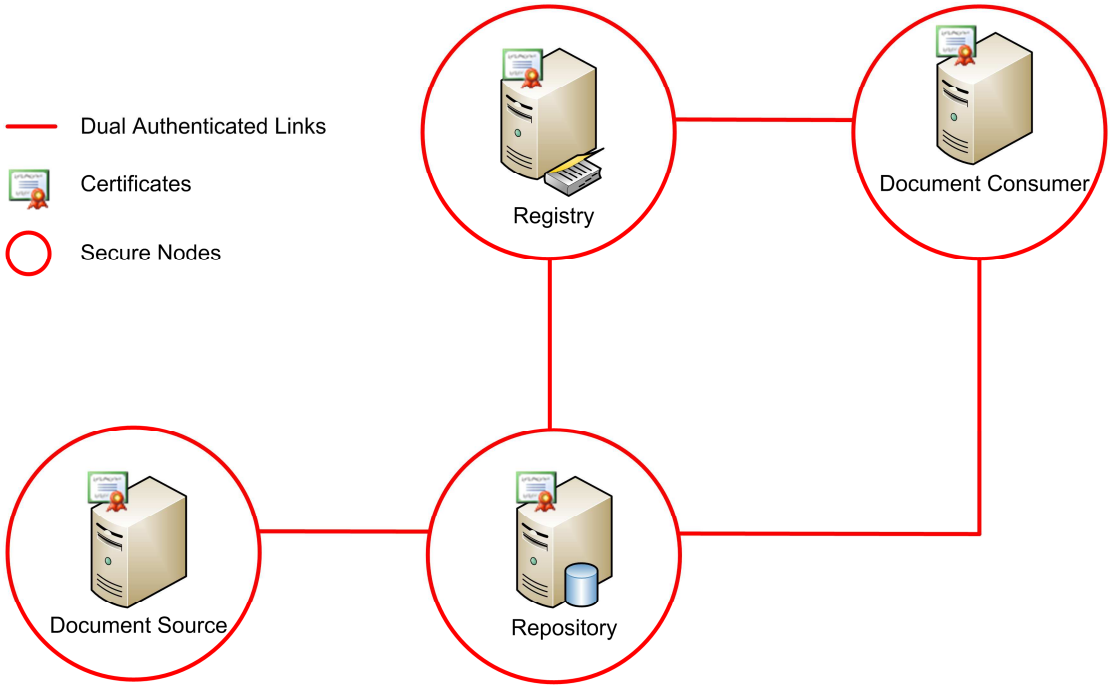


Image 17 : Authentification mutuelle des nœuds

Lors de chaque connexion entre Secure Nodes, le processus d'authentification doit avoir lieu pour vérifier l'identité de la machine. Ce système permet de s'assurer que les machines qui communiquent fassent bien partie du domaine d'affinité XDS. Dans le cas où un ordinateur arrivait à se connecter sur le réseau, puis à lancer une requête, celle-ci serait rejetée à cause de l'erreur d'authentification.

5.2 Schéma de la nouvelle architecture du prototype

Après cette vision théorique, l'architecture actuelle du prototype est prise en compte pour l'intégration des nouveaux profils. L'Image 18 représente l'architecture actuelle du prototype MediCoordination.

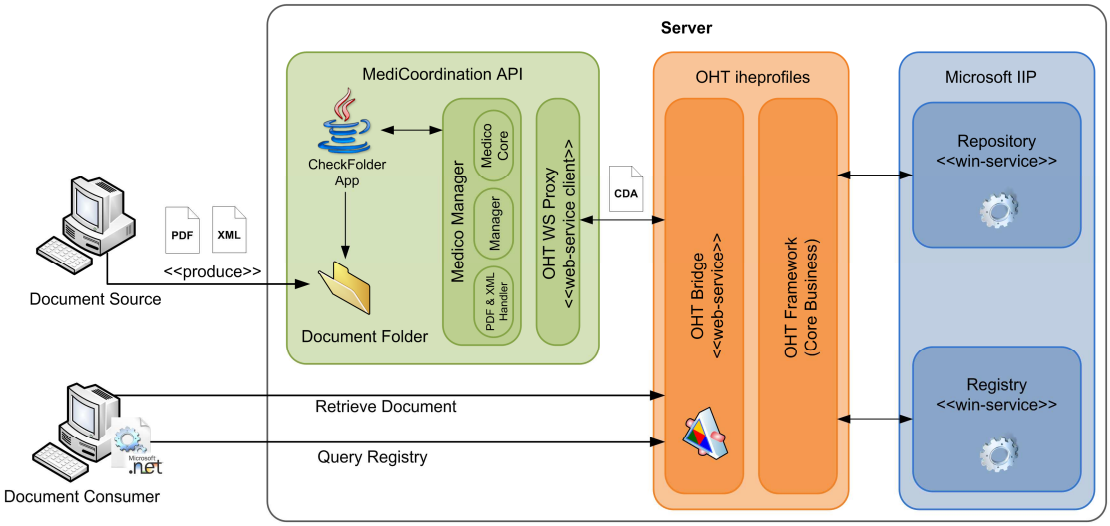


Image 18 : Schéma du prototype MediCoordination

Plusieurs éléments sont à retenir en ce qui concerne les différences entre le schéma théorique et l'architecture du prototype :

- Le web service joue le rôle d'intermédiaire entre les acteurs Document Source et Document Consumer et les acteurs Registry/Repository.
- L'acteur Document Source ne communique pas directement avec le web service. Les documents sont envoyés sur le serveur via FTP et c'est une application Java qui s'occupe de soumettre le document au web service.
- Les services Registry/Repository et le web service se situent actuellement sur le même serveur. Cependant, il faut prendre en compte que ce ne sera pas forcément le cas lors de l'intégration à une architecture déjà en place.
- L'équipe MediCoordination n'a pas un accès total aux acteurs Document Source et Document Consumer. Au niveau du client du médecin-privé, quelques modifications peuvent cependant être apportées au connecteur .Net.

Après avoir pris en compte ces remarques, les acteurs Secure Nodes sont redistribués de la façon suivante :

- Les communications de l'acteur Document Source vers le web service sont effectuées par le composant Java Médico Manager. Celui-ci devient alors le Secure Node à la place de l'ordinateur du RSV.
- Le Bridge est également un Secure Node puisque les requêtes des clients passent par lui.
- Les services Registry/Repository sont des Secure Nodes comme spécifié dans le schéma du profil.
- L'acteur Document Consumer n'est pas un Secure Node dans l'implémentation du prototype. Les développeurs de MediWay ne sont pas prêts pour prendre en charge le déploiement d'un certificat par machine. Dans le futur, il serait envisageable d'utiliser le certificat intégré à la carte d'assuré (type Swiss ID) pour le processus d'authentification.

Les authentifications mutuelles définies dans le profil ATNA sont ainsi déterminées via les Secure Nodes. Pour la gestion des certificats, deux choix sont envisageables :

- Validation des certificats basée sur la signature d'un CA
- Validation des certificats via des certificats autorisés

Après quelques recherches, le choix s'est porté sur la création d'une Autorité de Certification (CA) pour l'administration des certificats du prototype. Un certificat « root » est généré et a servi par la suite à signer les certificats des différents composants. Chaque machine a son propre certificat, ainsi que le certificat public du CA ce qui permet de valider automatiquement les certificats signés par celui-ci. Ainsi, cinq certificats sont générés par OpenSSL[28], qui est une boite à outils open source implémentant le protocole TLS, ainsi qu'une librairie pour la cryptographie.

Pour la seconde partie du profil ATNA, la gestion des logs est importante. L'OHT Bridge ainsi que les services Registry/Repository implémentent les transactions liées au profil. La base de donnée « Audit Record Repository » ainsi que le serveur de temps sont installés sur la même machine. Afin de déterminer quel système d'exploitation sera déployé sur le serveur, les dépendances de ces deux acteurs sont examinées. Au niveau du serveur NTP, celui-ci peut être configuré sous Linux, comme sous Windows. Pour le déploiement d'OpenATNA, ce projet nécessite l'installation d'un client Subversion, d'Apache Maven[29] et de Postgres[30]. Pour l'installation de ces derniers, une distribution Linux est préférable.

Subversion est un système de gestion de versions, distribué sous licence Apache et BSD (Berkeley Software Distribution). L'un des principes de Subversion est notamment le dépôt centralisé et unique des sources et les clients peuvent les télécharger et soumettre des modifications [44].

Apache Maven est un outil logiciel libre pour la gestion et l'automatisation de production de projets logiciels Java. L'objectif est de produire un logiciel à partir de ces sources, en optimisant les tâches réalisées à cette fin. Maven est géré par l'organisation Apache Software Foundation [45].

PostgresSQL est un système de gestion de base de données relationnelle et objet. C'est un outil libre, disponible selon les termes d'une licence de type BSD. Ce projet est fondé sur une communauté mondiale de développeurs et d'entreprises [46].

[28] http://www.openssl.org/
[29] http://maven.apache.org/
[30] http://www.postgresql.org/

La distribution Opensuse[31] a été retenue pour le système d'exploitation du serveur. Opensuse est une distribution Linux majeure, d'origine allemande et appartenant à la société américaine Novel. Elle est reconnue pour sa stabilité, sa gestion des paquets et son outil d'administration Yast. Cette implémentation est faite sur une machine virtuelle VMware[32] dans un environnement de test.

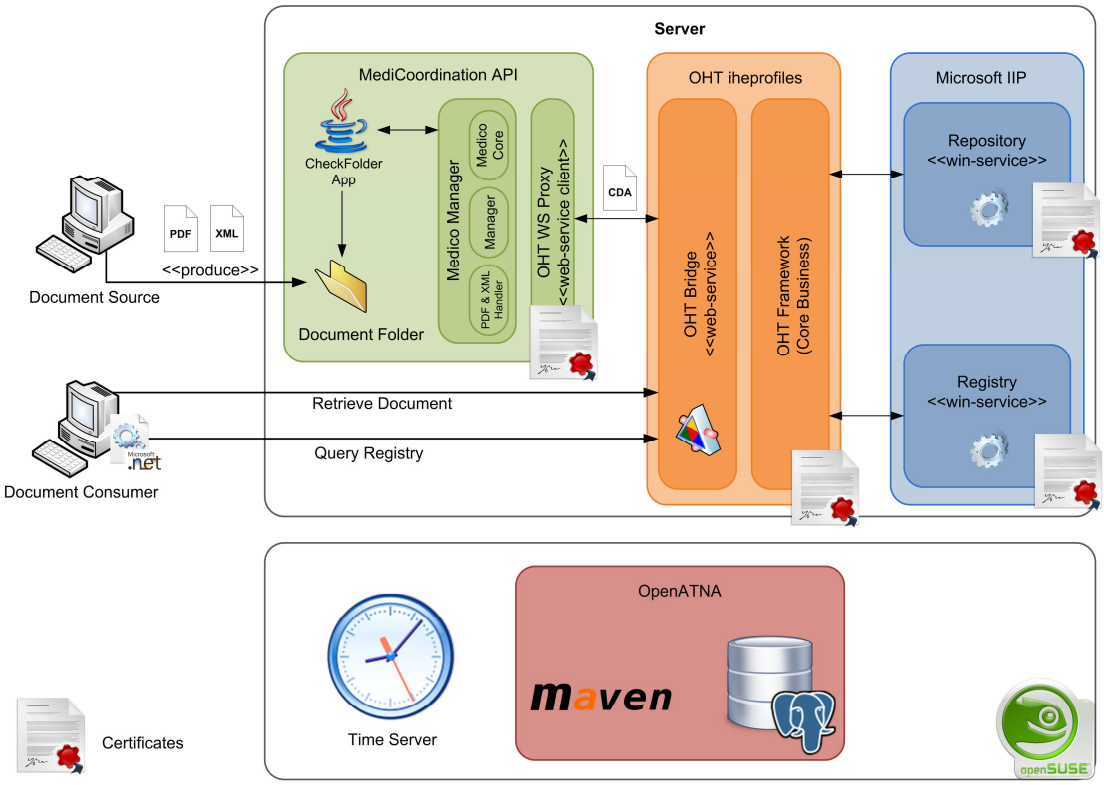


Image 19 : Architecture du prototype intégrant le profil ATNA

L'architecture du prototype MediCoordination intégrant le profil ATNA est représentée par l'Image 19. Un serveur dédié est mis en place pour le serveur NTP et la base données des logs. Au niveau du premier serveur, des certificats sont intégrés afin de permettre une authentification mutuelle des nœuds.

[31] http://www.opensuse.org/
[32] http://www.vmware.com/

6 Implémentation

Dans cette partie, les profils ATNA et CT sont implémentés et intégrés au prototype. L'implémentation se décompose en trois parts :

- Mise en place du serveur NTP et configuration des clients
- Installation de la base de données des logs
- Génération des certificats et intégration au prototype

Chaque point est développé ci-dessous. Les concepts généraux sont présentés et les différentes étapes sont décrites en détail.

6.1 Mise en place du serveur NTP et configuration des clients

La première chose à faire est la création d'une machine virtuelle. La distribution choisie est Opensuse 11.2 version 32 bits comme expliqué dans la section 5.2. Des mesures de sécurité habituelles sont effectuées, telles que le choix d'un mot de passe différent pour l'utilisateur par rapport au super utilisateur (root) et l'installation des dernières mises à jour.

Dans un second temps, un serveur NTP est installé. Pour rappel, NTP est un protocole qui permet de synchroniser l'heure d'un ordinateur avec celle d'un serveur de référence. Il rend possible la corrélation des événements de plusieurs systèmes. NTP utilise UDP (User Datagram Protocol) sur le port 123 pour recevoir les requêtes. Le fait d'employer un serveur NTP dans un réseau pour synchroniser les autres machines permet de réduire la consommation de bande passante dû au trafic NTP.

Sous linux, NTP est implémenté par le service ntpq. Sur la machine Opensuse, le paquet ntpq version 4.2.4 est installé par défaut avec un petit plugin pour Yast. Il est ainsi possible de configurer via une interface graphique le service. Il est également possible d'éditer directement le fichier `/etc/ntp.conf`.

Le service est configuré par le biais de ce fichier. La liste des serveurs NTP et les règles de sécurité y sont définies. Pour les aspects de sécurité, deux sections sont disponibles :

- Access Control : On peut définir les accès d'une machine ou d'une plage d'adresses IP au serveur. Il est donc possible de bloquer certaines adresses, ainsi que de choisir les droits d'une machine sur le serveur. Une liste des différentes actions est consultable dans la documentation [47].
- Authentication : Il est possible de demander une authentification du serveur NTP afin de valider le changement de temps. Plusieurs possibilités existent pour la mise en place de ce système, comme les clés symétriques, les clés publiques et autres [48].

Après la configuration du service via la mise à jour du fichier, une règle pour autoriser le protocole NTP est ajoutée au firewall (UDP port 123) et le service ntpq est configuré pour se lancer au démarrage de la machine. Un extrait du fichier de configuration est visible dans le Tableau 2.

```
…
authenticate no

server 0.pool.ntp.org iburst
server 1.pool.ntp.org iburst
server 2.pool.ntp.org iburst

#by default, ignore all ntp packets
restrict 0.0.0.0 mask 0.0.0.0 ignore

#restrict server permission
restrict 0.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
restrict 1.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
restrict 2.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery

#allow client query from lab network
restrict 192.168.169.0 mask 255.255.255.0 nomodify notrap
…
```

Tableau 2 : Extrait du fichier ntp.conf

Dans l'extrait du fichier ntp.conf, l'authentification des paquets est désactivée. Ensuite, le « pool » est spécifié pour la synchronisation du temps et enfin, les contrôles d'accès sont définis.

L'authentification n'est pas mise en place afin de gagner du temps et les recommandations IHE précisent que son implémentation n'est pas justifiée vu le coût de maintenance et le risque très faible de piratage à cet endroit du système. De plus, l'implémentation cliente du protocole NTP par Microsoft ne permet pas l'authentification, un outil tiers serait donc requis.

Le serveur NTP local est configuré pour se synchroniser avec des serveurs NTP distants. Le « pool.ntp.org[33] » est un projet regroupant un grand nombre de serveurs NTP afin de fournir le temps à des millions de clients.

Pour ce qui est du contrôle d'accès, le serveur est libre d'accès si rien n'est spécifié dans le fichier de configuration. Les lignes suivantes indiquent que tous les paquets NTP sont ignorés par défaut, que les serveurs sont autorisés à communiquer pour modifier l'heure, mais ne sont pas autorisés à interroger ou modifier les

[33] http://www.pool.ntp.org/en/

configurations du serveur et que les machines du réseau local sont autorisées à interroger le serveur.

Le serveur étant prêt, il ne reste plus qu'à configurer les clients. Dans notre implémentation, il n'y a qu'un seul client, le serveur MediCoordination tournant sur Windows XP. Pour configurer un client sous le système d'exploitation de Microsoft, il faut aller dans Démarrer / Panneau de configuration / Date et heure / onglet Temps Internet et insérer l'adresse IP du serveur NTP.

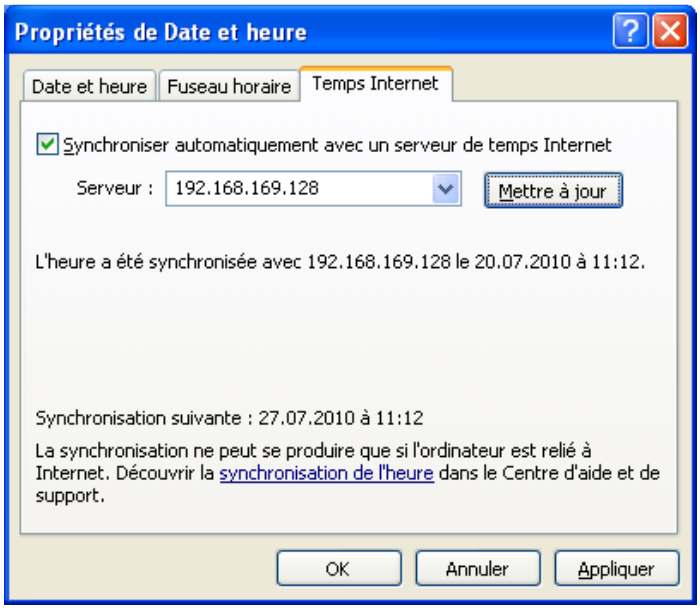


Image 20 : Synchronisation du client NTP

Le client est maintenant à la même heure que le serveur NTP. Lorsque les différents composants du prototype envoient des logs à l'audit record repository, il y aura une cohérence entre les événements.

Afin de bien configurer le service NTP, plusieurs tutoriels ont été lus, ainsi que la documentation officielle sur ntp.org [49] [50] [51].

6.2 Installation de la base de données des logs

La deuxième phase de l'implémentation consiste à mettre en place l'acteur Audit Record Respository. L'outil choisi est OpenATNA d'OpenHealthTools. Plusieurs outils sont installés à cet effet via le gestionnaire de paquets Yast :

- Subversion 1.6.6
- PostgresSQL 8.4.4
- Maven 2.2.1
- Java 1.6

La première étape pour l'installation du projet OpenATNA est la configuration de PostgreSQL. L'installation du package `postgres-server` via Yast ajoute un utilisateur nommé « postgres » qui sera le DBA (Database Administrator) du serveur postgres. Un script de gestion du service est copié dans `/etc/init.d/postges`. Les fichiers de postgres se situent dans `/var/lib/pgsql/`, dont les plus importants sont le dossier data contenant les bases de données et le fichier pg_hba.conf correspondant à la gestion des authentifications.

Dans la plupart des installations, ce fichier contient la valeur « trust » pour les utilisateurs locaux, ce qui signifie que tous les utilisateurs du système peuvent se connecter au service. Dans le cas de l'installation par Yast, c'est la valeur « ident » qui est en place. Lors d'une tentative de connexion, le serveur récupère le nom de l'utilisateur via le système d'exploitation et vérifie qu'il correspond à un utilisateur de postgres. Si c'est le cas, l'utilisateur est autorisé à se connecter et dans le cas contraire, un message d'erreur est affiché. D'autres choix sont possibles pour l'implémentation de la gestion des autorisations, tels qu'un mot de passe en clair, un hash md5, Kerberos et des certificats [52]. La modification principale apportée à ce fichier est l'interdiction de se connecter à distance, ce qui réduit les risques d'intrusion et correspond aux besoins du prototype, puisque les connexions à la base de données par les autres composants s'effectuent via le service Java. Dans un second temps, il a fallu changer le moyen d'authentification car le mode « ident » n'est pas supporté par le client Java. Le hash md5 a été choisi comme remplaçant.

Le fichier d'autorisation spécifiant « ident » pour les utilisateurs locaux et ne connaissant que l'utilisateur « postges », le seul moyen de s'authentifier est de se loguer en tant que postgres en passant par root. Après s'être connecté au serveur, le mot de passe du compte postgres a été modifié et le fichier pg_hba.conf a été rectifié.

Ensuite, l'utilisateur « openatna » est ajouté avec un bon mot de passe, puis une base de données appelée également openatna est créée et associée à l'utilisateur openatna. Le serveur étant prêt, il est configuré via le module runlevel de Yast pour se lancer au démarrage du système.

La seconde étape consiste à télécharger les sources du projet OpenExchange et OpenATNA grâce au client subversion. Le répertoire choisi pour stocker les fichiers est `/srv`, place réservée sous unix pour les services web et ftp. La commande pour récupérer les fichiers sources est présentée dans le Tableau 3.

```
svn checkout
https://openatna.projects.openhealthtools.org/svn/openatna/trunk
openatna --username username
```

Tableau 3 : Récupération des sources du projet OpenATNA

Avant de compiler et d'installer le service Java d'OpenATNA, il faut modifier le fichier `openatna/audit/src/main/resources/openatna.properties` pour y inclure les crédentiels d'accès à la base de données, ainsi que le fichier ArrConnections.xml afin que le service n'écoute pas seulement en local.

Cette modification faite, les sources sont compilées via l'utilitaire maven 2. Plusieurs fichiers sont créés dans le repository local de maven lors de la compilation, dans mon cas `/root/.m2/repository`. Plusieurs manipulations sont disponibles grâce à maven, dans le cas du Tableau 4, un exécutable (.jar) est créé dans le dossier `/all/build/`, ainsi qu'un dossier lib contenant toutes les dépendances.

```
mvn –P exec install
```

Tableau 4 : Création d'un exécutable d'OpenATNA

Afin de lancer plus facilement l'exécutable, un script a été créé. Il suffit de taper à la console « openatna » pour démarrer le client Java. Le serveur lancé, deux ports sont accessibles d'Internet :

- TLS sur le port 2862
- UDP sur le port 2863

Le serveur Linux est dès à présent prêt à l'emploi. Il faut maintenant configurer les clients pour envoyer leurs logs à ce serveur. Pour ce faire, il faut effectuer les modifications suivantes :

- Pour les services wcf, il faut modifier dans la balise <appSettings> les valeurs suivantes : `<add key="SYSLOG_SERVER" value="192.168.169.128" />` et `<add key="SYSLOG_PORT" value="2863" />` dans les fichiers `App.config` du Registry et du Repository.
- Pour le bridge, il faut modifier le fichier rhioConfig.xml dans le répertoire de configuration de tomcat. Le fichier RHIO (Regional Health Information Organization) est une liste des différents acteurs IHE (XDS registries, repositories, PIX managers, ATNA audit record repositories) et leurs informations de configuration associées. Ainsi, le client peut identifier l'acteur qu'il veut contacter par son nom et ne se soucie pas de la connectivité.

Afin de prendre ces nouvelles informations en compte, il faut relancer le serveur tomcat et réinstaller les services Windows. La gestion des logs est maintenant entièrement implémentée et opérationnelle. Dans le but d'interroger le serveur des logs, un servelet (page web Java) est ajouté dans le répertoire « web » d'OpenATNA. Lors de la compilation, un fichier .war a été créé et peut être installé dans un conteneur comme tomcat. Pour un simple test, il est possible d'utiliser le serveur web Jetty[34] en tapant la commande suivante dans le dossier web :

```
mvn jetty:run
```

Tableau 5 : Lancer le servelet dans un conteneur jetty

Une fois Jetty démarré, une page web est accessible à l'adresse : http://localhost:8081/atna/atna. Il est alors possible d'interroger la base de données. Deux sections sont disponibles : les erreurs et les messages. Après avoir choisi une section, la page permet de filtrer le contenu selon plusieurs critères. Un exemple de messages ATNA est visible dans l'Image 21.

Event Type Code	***Code System***	***Code System Name***	***Display Name***
ITI-41		IHE Transactions	Provide and Register Document Set-b
Source ID		**Enterprise Site ID**	
Arnaud			
User ID		**Alt User ID**	**User Name**
http://www.w3.org/2005/08/addressing/anonymous		1636	
Participant Type Code	***Code System***	***Code System Name***	***Display Name***
110153		DCM	Source
Network Access Point ID		***Type***	
192.168.169.129		2	
User ID		**Alt User ID**	**User Name**
https://hesso-mc-gw:8013/XdsService/XdsRepository			
Participant Type Code	***Code System***	***Code System Name***	***Display Name***
110152		DCM	Destination
Network Access Point ID		***Type***	
hesso-mc-gw		1	
Object ID		**Object Name**	
498ef443e7ac4a6^^^&1.3.6.1.4.1.21367.2005.3.7&ISO			
Object ID Type Code	***Code System***	***Code System Name***	***Display Name***
2		RFC-3881	Patient Number
Object Type Code	**Object Type Code Role**	**Object Sensitivity**	
1	1		
Object ID		**Object Name**	
1280913684662			
Object ID Type Code	***Code System***	***Code System Name***	***Display Name***
urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd		IHE XDS Metadata	SubmissionSet ClassificationNode
Object Type Code	**Object Type Code Role**	**Object Sensitivity**	
2	20		

Image 21 : Exemple d'un message ATNA

Tous ces champs ci-dessus sont définis de manière précise dans la RFC3881. Un administrateur peut consulter les logs de manière très précise via cette page web et ainsi ajuster les règles de sécurité en cas d'incohérence.

[34] http://jetty.codehaus.org/jetty/

Comme dit précédemment, deux protocoles sont disponibles dans l'implémentation d'OpenATNA : TLS et BSD Syslog. Dans le cas de notre prototype, c'est le deuxième qui est employé pour la simple raison que les composants déjà en place (bridge et services wcf) ne supportent que le protocole BSD Syslog.

6.3 Génération des certificats et intégration au prototype

L'acteur audit record repository et les clients étant configurés, il ne reste plus que l'authentification des nœuds pour finaliser l'implémentation du profil ATNA. Pour ce faire, plusieurs certificats sont générés pour les différents composants choisis dans la section 5.2.

Pour rappel, le but premier des certificats est de vérifier l'origine de données signées, comme un email. Il est alors possible pour le récepteur de connaître l'origine des données et si elles ont été altérées durant le trajet. Les informations les plus importantes d'un certificat sont le « distinguished name » (DN) et une clé publique. Le DN est l'identifiant d'une entité, dans notre cas un composant, qui possède la clé privée qui correspond à la clé publique du certificat. De manière générale, la clé privée est utilisée pour crypter les données et la clé publique du certificat sert à les décrypter. Un certificat signé par la clé privée qui correspond à la clé publique du certificat est appelé un certificat autosigné (self-signed certificate). Le certificat d'une autorité de certification (CA) tombe dans cette catégorie. Les certificats des machines sont souvent signés par une clé privée différente, comme une clé privée d'un CA. Cela constitue une chaine de deux certificats (chain trust). Ainsi, le certificat de la machine est validé car il est signé par une autorité de confiance, le CA.

Dans le cadre de cette implémentation, les certificats générés correspondent à la représentation de l'Image 22.

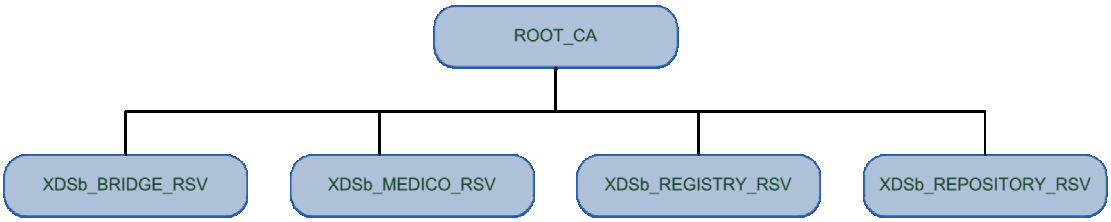


Image 22 : Architecture des certificats

6.3.1 Création des certificats

Il existe plusieurs outils pour la création de certificats. Il y a notamment keytool, un utilitaire en ligne de commande livré avec le SDK de Java. Cependant, pour une gestion plus complexe des certificats, il nous faut un logiciel plus complet tel qu'OpenSSL [53].

La première chose à faire est d'installer OpenSSL. Celui-ci est, en général, déjà installé sur les systèmes d'exploitations linux. Pour Windows, il suffit de le télécharger sur SourceForge.net[35]. En plus des binaires d'OpenSSL, il faut également avoir le scripte CA.sh qui peut être téléchargé dans les sources[36].

6.3.1.1 Création d'un certificat root

Le scripte CA.sh est employé pour créer plus facilement un certificat root. En fait, ce script utilise des commandes d'OpenSSL pour générer les clés nécessaires et effectuer les opérations de signature. La première commande à exécuter est :
`CA -newca`.

Plusieurs informations sont demandées lors de la génération du certificat. A la fin du processus, une clé privée est créée, ainsi qu'un certificat contenant la clé publique. Le certificat root peut être regardé dans le Tableau 6. On y aperçoit notamment la contrainte CA:true.

Les éléments du CA étant créé, il est possible de générer les certificats pour les différents composants.

[35] http://gnuwin32.sourceforge.net/packages/openssl.htm
[36] http://www.openssl.org/source

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            c4:eb:35:5a:e9:ca:33:5c
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=CH, ST=Valais, O=HES-SO,
CN=RSV_ROOT_CA/emailAddress=gaspoz.arnaud@gmail.com
        Validity
            Not Before: Jul 27 16:40:30 2010 GMT
            Not After : Jul 26 16:40:30 2013 GMT
        Subject: C=CH, ST=Valais, O=HES-SO,
CN=RSV_ROOT_CA/emailAddress=gaspoz.arnaud@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c8:76:d9:6e:69:19:b7:7f:27:63:c9:8a:04:3c:
                    a9:b0:c3:b5:fb:9c:b0:a6:88:67:0d:72:56:02:4e:
                    a7:96:f6:4a:70:da:ba:66:c9:9b:bf:30:5c:58:e2:
                    6c:16:60:b9:a7:02:4b:8b:bb:e8:56:38:df:cf:51:
                    6e:72:ed:6a:7d:fa:1b:62:e8:b2:da:6f:47:65:f1:
                    eb:b5:d4:b2:fb:e7:b7:17:cc:ed:0a:49:42:ae:bf:
                    d0:bc:02:74:71:69:05:a1:3b:0f:f1:f9:6e:ec:89:
                    20:c3:4e:8d:5f:6e:68:a5:28:da:c4:82:00:65:63:
                    22:18:9f:38:36:4f:59:2e:49
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:

51:A4:DC:38:06:04:96:DE:C4:DC:31:3E:FA:45:4C:0F:45:AE:5F:5A
            X509v3 Authority Key Identifier:

keyid:51:A4:DC:38:06:04:96:DE:C4:DC:31:3E:FA:45:4C:0F:45:AE:5F:5A
                DirName:/C=CH/ST=Valais/O=HES-
SO/CN=RSV_ROOT_CA/emailAddress=gaspoz.arnaud@gmail.com
                serial:C4:EB:35:5A:E9:CA:33:5C

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha1WithRSAEncryption
        a9:84:c0:e3:ab:74:be:53:12:e7:23:bc:65:55:b5:4d:23:ca:
        c0:93:af:7d:84:2c:3a:5b:6d:5e:cd:78:41:66:18:cb:a5:7a:
        34:a7:c0:ce:6d:4b:12:90:44:90:7d:43:2e:c9:94:b2:a2:4b:
        0d:2a:d5:55:ee:d3:41:78:27:f7:9a:0b:2d:9f:33:f7:85:43:
        92:33:2c:7e:b4:e7:f3:27:59:bb:b7:11:56:1e:a0:f9:22:be:
        b1:6e:cd:d1:d2:a5:cd:92:97:d3:e8:cf:c0:f5:df:68:33:3c:
        e2:f7:ea:2f:01:c2:8b:0e:d0:e7:bc:c6:f2:8f:d8:72:70:ce:
        92:bc
```

Tableau 6 : Certificat du CA

6.3.1.2 Générer les certificats pour les composants

Le processus de création des seconds certificats se passe en deux étapes : générer une requête et signer la requête. Le script CA.sh est utilisé dans les deux cas.

La génération de la requête est faite par le biais de la commande `CA -newreq`, ce qui initialise le même dialogue que celui de la création du certificat root. A la fin de l'exécution du script, un fichier nommé newreq.pem est créé et est considéré comme un certificat pas encore signé.

L'exécution de la commande `CA -sign` signe la requête en utilisant la clé privée du CA. A ce stade, un certificat signé par le CA et une clé privée sont disponibles, ainsi que ceux du CA. Le certificat pour le bridge est visible dans le Tableau 7.

Tous les certificats sont maintenant générés et il ne reste plus qu'à les mettre en place dans les différents composants du prototype. Pour rappel, l'application MediCo API et le bridge sont écrits en Java, alors que les services Registry / Repository s'appuient sur le framework .Net. Par conséquent, les deux services wcf ne peuvent être installés que sur un système d'exploitation Windows, alors que Java tourne sur toutes les plateformes. Il est donc important de comprendre que ces deux systèmes ne gèrent pas les certificats de la même manière.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            c4:eb:35:5a:e9:ca:33:5d
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=CH, ST=Valais, O=HES-SO,
CN=RSV_ROOT_CA/emailAddress=gaspoz.arnaud@gmail.com
        Validity
            Not Before: Jul 27 16:42:31 2010 GMT
            Not After : Jul 27 16:42:31 2011 GMT
        Subject: C=CH, ST=Valais, L=Sierre, O=HES-SO,
CN=XDSb_BRIDGE_RSV/emailAddress=gaspoz.arnaud@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:de:bf:d7:ad:b2:9e:c9:0b:63:4e:61:2e:d5:48:
                    7e:80:d9:c2:c2:e7:7e:df:b6:1d:57:e9:23:b0:87:
                    3f:31:de:5e:7c:e3:eb:e8:e9:b1:7f:1a:a9:ff:4f:
                    f8:fd:83:7b:26:9e:40:5f:51:62:0a:2c:f7:be:fa:
                    56:c3:e5:09:87:84:f3:9a:5d:fd:45:bb:5e:77:a4:
                    d6:da:14:32:21:48:86:8a:d8:f2:97:4c:3d:be:92:
                    02:30:1c:16:5a:61:93:78:36:54:a1:21:56:c1:b0:
                    c2:98:0f:75:09:29:69:f4:6b:2e:16:b4:10:74:c3:
                    f7:5c:ca:13:bc:11:20:1c:47
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:

7E:EA:A3:AF:7E:95:97:BD:E6:39:DA:D7:F7:07:1A:D2:CD:4F:4F:35
            X509v3 Authority Key Identifier:

keyid:51:A4:DC:38:06:04:96:DE:C4:DC:31:3E:FA:45:4C:0F:45:AE:5F:5A

    Signature Algorithm: sha1WithRSAEncryption
        94:df:c8:b0:c6:9d:6b:ee:80:90:9e:17:1a:d8:3f:49:d6:20:
        fa:d4:ce:26:3e:88:cc:cc:d7:3e:b8:1b:a2:a4:dc:1c:9a:ee:
        d9:cf:47:69:8d:f4:81:47:a7:43:c2:ce:12:d5:8a:79:7b:22:
        61:8e:93:5c:2a:a6:fa:7b:9e:f2:b3:34:a7:07:93:dd:1d:94:
        3e:8d:f0:68:ab:be:eb:3b:1a:ef:f2:64:a7:32:6a:fb:a0:e9:
        f7:dc:4a:9d:cc:d3:90:c2:3f:77:7e:b9:18:e8:df:1b:b3:49:
        ae:17:11:87:12:4f:e9:96:56:d8:2c:8e:43:64:56:53:bd:52:
        f6:a7
```

Tableau 7 : Certificat du bridge

Les applications .Net s'appuient sur la gestion des certificats de Windows. Ainsi, les certificats sont placés dans la console des certificats de la machine sous les onglets Personnel pour les composants et Autorité de certification racine de confiance pour le certificat root.

En ce qui concerne les applications Java, elles ne peuvent pas utiliser une fonctionnalité du système d'exploitation, puisque celui-ci peut être différent. Java utilise à la place des keystores et truststores. Un keystore est un conteneur dans lequel se trouve une clé privée pour l'encryptage des données et le certificat contenant la clé publique correspondante. Le truststore quant à lui contient les certificats des serveurs avec lesquels l'application communique.

L'extension des keystores et truststores est .jks et ce format de fichier ne peut pas être généré par OpenSSL. Il faut donc passé par un intermédiaire, le format pkcs12. PKCS (Public Key Cryptographic Standards) numéro 12 définit un format de fichier généralement utilisé pour stocker la clé privée et le certificat contenant la clé publique correspondant en les protégeant par un mot de passe [54].

OpenSSL permet de générer un conteneur pkcs12 avec la commande suivante :

```
openssl pkcs12 -export -out bridge_keystore.pkcs12 -in
bridge.cert.pem -inkey bridge.key.pem
```

Dès que les conteneurs pkcs12 sont créés, il suffit de placer le certificat root dans les autorités de certificat racine de confiance et les fichiers repository.pkcs12 et registry.pkcs12 dans l'onglet Personnel des certificats de la machine. Pour les applications Java, les conteneurs pkcs12 sont convertis en .jks via la commande suivante :

```
java -cp c:\jetty\lib\jetty-6.1.1.jar
org.mortbay.jetty.security.PKCS12Import bridge_keystore.pkcs12
bridge_keystore.jks
```

Pour configurer l'authentification mutuelle du bridge, il faut modifier le fichier de configuration principal bridge.properties, en modifiant les valeurs suivantes :

- javax.net.ssl.keyStore
- javax.net.ssl.keyStorePassword
- javax.net.ssl.trustStore
- javax.net.ssl.trustStorePassword

Afin de configurer l'application MediCo API, il faut modifier les entrées de la classe Configuration et y insérer les informations relatives au keystore et truststore.

Les modifications à apporter aux services wcf s'effectuent dans les fichiers app.config du registry et du repository. Il faut ajouter le code suivant dans les comportements du service.

```
<serviceCredentials>
      <serviceCertificate
            findValue="XDSb_REGISTRY_RSV"
            storeLocation="LocalMachine"
            storeName="My"
            x509FindType="FindBySubjectName" />
      <clientCertificate>
            <authentication
                  certificateValidationMode="ChainTrust"
                  revocationMode="Offline" />
      </clientCertificate>
</serviceCredentials>
```

Tableau 8 : Modification pour valider un certificat via Chain Trust

En plus d'avoir modifié ces deux fichiers de configuration, il faut également, dans le cadre d'un service wcf auto-hébergé via la classe WSHttpBinding utilisant la sécurité de transport, configurer le port avec un certificat X.509. Pour se faire, l'utilitaire httpcfg est employé puisque le système d'exploitation en place est Windows XP [55].

La mise en place des certificats est maintenant terminée. La dernière chose à modifier est la notion de SessionContext. Le bridge est un service sans état (stateless), ce qui signifie que le web service n'a pas de mémoire entre deux invocations. Afin de faciliter la configuration et l'utilisation du bridge, quelques paramètres sont transmis par l'utilisateur à travers un objet SessionContext.

Les principales informations à spécifier dans le SessionContext sont le rhioName, useSecuredConnectionWhenAvailable et auditSourceId. Pour rappel, l'indication rhio permet d'identifier la bonne infrastructure IHE. La seconde variable précise au bridge d'utiliser ou non les urls sécurisées, si disponibles. Enfin, l'audiSourceId est l'identifiant employé dans les logs afin de déterminer quel système est impliqué [56]. Ces changements sont présents dans le Tableau 9.

```
sessionContext.setRhioName("XDS.b RSV");
sessionContext.setUseSecuredConnectionWhenAvailable(true);
sessionContext.setAuditSourceId("Arnaud");
```

Tableau 9 : Modification du SessionContext

Les différents composants du prototype étant des services, il est alors difficile de démontrer l'utilisation de TLS. Puisque tous les services se situent sur la même machine, il n'est pas possible d'utiliser un analyseur de protocoles tel que Wireshark[37] pour voir les trames, puisque celles-ci ne sont jamais créées. Le seul moyen de vérifier l'emploi des certificats est de consulter les logs. Les lignes les plus intéressantes sont reprises dans le Tableau 10.

```
…
System property javax.net.ssl.keyStore set to
conf/keystores/bridge_keystore.jks
System property javax.net.ssl.keyStorePassword set to XXX (password
not shown)
System property javax.net.ssl.trustStore set to
conf/keystores/bridge_truststore.jks
System property javax.net.ssl.trustStorePassword set to XXX
(password not shown)
…
Connection succesfully made using TLS to host hesso-mc-gw on port
8013
…
```

Tableau 10 : Logs reportant l'utilisation de TLS

Le profil ATNA est maintenant entièrement implémenté. Cependant, plusieurs remarques et améliorations sont décrites dans la section suivante.

6.4 Améliorations possibles

Les deux profils sont maintenant implémentés dans le prototype. Cependant, plusieurs améliorations peuvent être apportées. La première remarque est que l'application Java MediCo API ne génère pas de logs contrairement à ce qui est exigé par le profil ATNA pour un secure node. Il est possible de télécharger les classes Java pour le format des logs définis dans la RFC-3881. Il faudrait modifier le code afin de générer des logs ayant ce format et les envoyer via UDP à l'acteur Audit Record Repository. Une contrainte est, comme discuté dans la section 3.3.2, qu'il n'y pas de notion d'utilisateur lors de la connexion de l'application Java vers le bridge. Il est alors impossible pour le moment de générer des logs quand tel utilisateur dépose tel fichier.

[37] http://www.wireshark.org/

Une autre faiblesse du prototype est l'impossibilité pour le moment d'ajouter un certificat par client (MediWay). Les communications entre les médecins-privé et le bridge ne sont pas authentifiées de manière mutuelle. L'utilisation des certificats intégrés aux cartes d'assurés permettrait une authentification mutuelle simplifiée et sécuriserait ainsi cette partie critique.

Actuellement, les connexions entre le Repository et le Registry se font via HTTP et ne sont pas mutuellement authentifiées. La raison est que les applications .Net sont plus strictes pour la vérification des certificats que Java. Le Distinguished Name est comparé au nom de la machine et s'il ne correspond pas, la connexion est refusée. Les acteurs Registry / Repository étant sur la même machine, les certificats ne sont pas bien nommés et posent donc problème. Cette situation sera résolue lorsque les services seront sur des machines différentes, ceci est le cas dans une implémentation concrète, représentée par l'Image 15.

Une dernière chose à vérifier est la génération des logs. Le bridge et IIP spécifient être conformes au profil ATNA et donc envoient les logs exigés à la base de données. Cependant, il est recommandé de vérifier que les applications génèrent bien tous les logs et dans le cas contraire, compléter les codes sources en conséquence.

7 Conclusion

Tous les aspects décrits dans le cahier des charges ont été développés dans ce dossier. Après avoir lu les recommandations faites dans l'audit de sécurité, une analyse des profils IHE a été effectuée afin de déterminer si IHE propose des solutions pour sécuriser les aspects relevés précédemment. Par la suite, les profils IHE ATNA et CT ont été choisis et implémentés dans l'architecture actuelle.

Ces profils ont permis l'intégration de la traçabilité médico-légale, ainsi que l'authentification mutuelle des nœuds. Les machines sont donc identifiées comme appartenant au réseau et les communications sont sécurisées via TLS. Un administrateur peut dorénavant consulter les logs des différents composants de manière centralisée et prendre les décisions qui conviennent. De plus, le prototype se rapproche encore plus des spécifications IHE afin d'être conforme aux exigences du profil XDS.

De manière globale, les objectifs ont été atteints dans le temps impartis, il faut cependant prendre en compte les remarques formulées dans la section 6.4. Ce projet a ainsi contribué à renforcer la sécurité du prototype.

8 Gestion de projet

Dans cette partie, le travail de bachelor est revu dans une optique de gestion de projet.

8.1 Déroulement

Ce travail a été effectué en 371 heures réparties sur 13 semaines. Les 4 premières semaines se sont passées en parallèle avec les cours, à raison de 20 heures semaines. Ensuite, 2 semaines étaient réservées pour la préparation et les examens, mais ont été malgré tout utilisées pour le projet. Enfin, 7 semaines à 45h se sont enchainées jusqu'à la fin du travail le 16 août 2010.

Durant les deux premières semaines, l'analyse du projet a permis la rédaction du cahier des charges et de la planification. En parallèle, des recherches ont été effectuées pour avoir une vue globale du sujet.

Deux semaines ont été ensuite employées pour la lecture des différents documents de base. La rédaction du dossier a débuté et la plupart des tâches administratives ont été mises en place. Il a été choisi d'écrire le dossier tout au long du projet afin de ne pas oublier de détails.

La partie « Comprendre l'architecture actuelle » a été complètement rédigée durant les deux semaines suivantes et l'audit de sécurité partiellement terminée. M. Schumacher partant en vacances le 08 juillet, une proposition d'implémentation, ainsi que le design du prototype devaient lui être présentés. C'est pourquoi, l'audit de sécurité a été mis en attente et l'analyse des profils IHE a débuté.

L'analyse des profils IHE a été réalisée en 1 semaine et la conception du nouveau prototype a été formulée. La semaine d'après, M. Schumacher a validé mes propositions et la deuxième partie de l'audit de sécurité a été ajoutée.

Les trois semaines d'implémentation ont alors débuté. En passant de la création de la machine virtuelle à la mise en place des différents composants, plusieurs problèmes ont dû être résolus pour faire fonctionner le prototype.

Enfin, les deux dernières semaines ont été consacrées à la finalisation du dossier et à sa relecture.

8.2 Planification

Selon la planification initiale disponible en annexe et reproduite ci-dessous, les délais ont été respectés et aucune adaptation n'a dû être effectuée. Les seules contraintes quant à l'échelonnage des heures sur la durée prévue étaient de prendre en compte que certaines personnes partaient en vacances, comme M. Alves en tout début de projet et M. Schumacher à partir du 08 juillet. Il fallait donc que la phase de design soit finie pour cette date afin de valider mes choix.

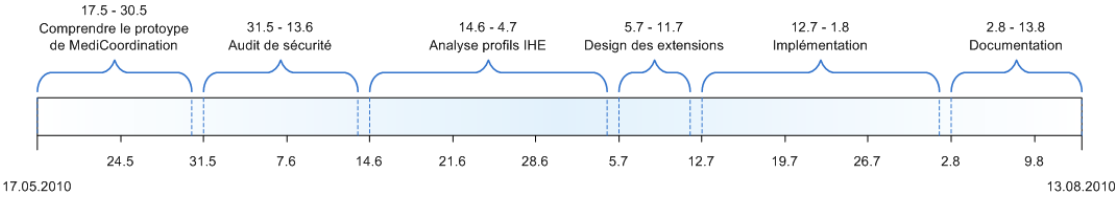


Image 23 : Planification initiale

Malgré une bonne planification et très peu de modifications, la partie « Comprendre l'architecture actuelle » a dépassé de quelques jours l'échéance en raison du temps accordé pour préparer les examens et la rédaction partielle du dossier. De plus, la fin de l'audit de sécurité a été rédigée après le design des extensions afin de pouvoir proposer une implémentation à temps.

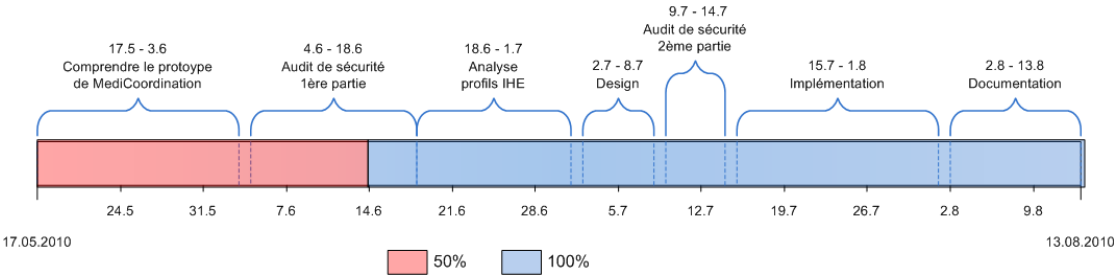


Image 24 : Planification finale

L'Image 24 présente la situation finale du déroulement du travail.

8.3 Suivi hebdomadaire

Durant tout le projet, des séances hebdomadaires ont été fixées afin de communiquer l'avancement du projet à mon responsable, valider certains choix et définir les prochains objectifs.

8.4 Bilan des heures effectuées

Dans l'Image 25, un comparatif des heures prévues par l'école par rapport aux heures effectuées est disponible. Malgré certains jours fériés durant le temps réservé pour le travail de bachelor, le nombre d'heures prévu en début de projet a pu être tenu.

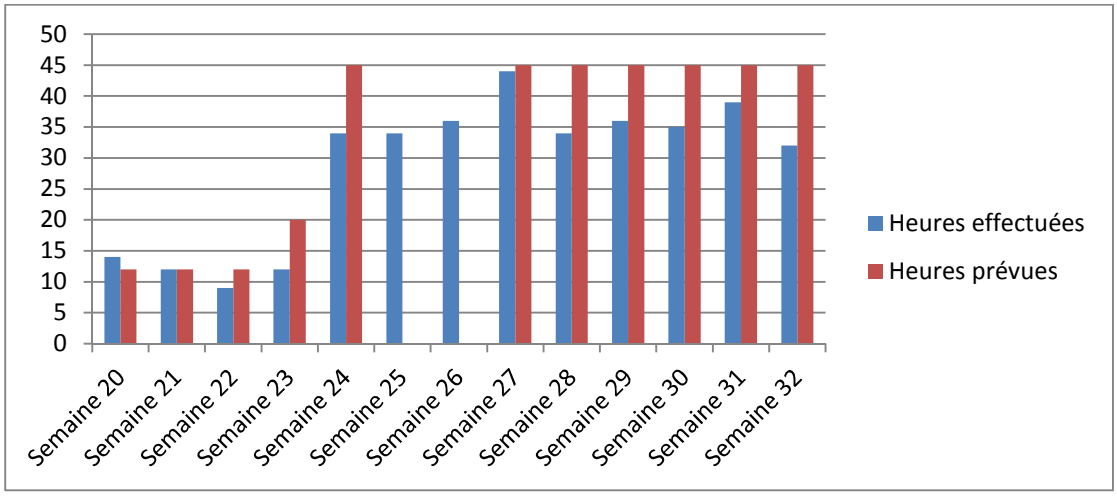


Image 25 : Bilan des heures effectuées

Lors des semaines 22 et 23, du temps a été pris pour la préparation des examens. Durant les semaines 25 et 26, j'ai travaillé à plein temps sur le projet puisque j'étais exempté d'examens de module. Ainsi, avec ce temps supplémentaire, les heures des semaines suivantes ont été réduites.

9 Satisfaction personnelle

Dès le départ, j'ai été motivé par le thème de ce projet. Le domaine de la sécurité informatique me passionnant, ce travail de bachelor m'a permis d'œuvrer sur un projet répondant à mes attentes.

Je me suis familiarisé avec la sécurité des web services, ainsi que plusieurs mécanismes de sécurité orientés XML. Les différents aspects de la sécurité d'une architecture m'ont été démontrés par le biais de l'audit de sécurité. L'intégration des certificats dans le prototype lors de la phase d'implémentation m'a donné une vision plus pratique que celle vue en classe.

Une grande liberté de mouvement m'ayant été accordée et une bonne entente au sein des membres du projet, l'ambiance de travail fut idéale. Ce travail de bachelor a été une agréable expérience dans son ensemble.

10 Remerciements

Je tiens à remercier toutes les personnes ayant contribué de près ou de loin à la réalisation de ce travail.

Je tiens à remercier particulièrement Monsieur Michael Schumacher pour avoir proposé un thème axé sur la sécurité informatique et pour m'avoir suivi et encadré tous le long du projet.

Un grand merci également à Monsieur Bruno Alves pour m'avoir soutenu lors de l'élaboration de ce travail de bachelor. Il m'a aidé à la prise en main du prototype et à régler certaines erreurs de configuration.

Merci aux personnes qui m'ont aidé à relire et à peaufiner ce document. Merci pour leurs efforts et leur patience.

11 Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- M. Schumacher, professeur à la HES-SO Valais
- M. Alves, assistant à la HES-SO Valais
- M. Buri, responsable de la sécurité informatique au RSV de Sion

Sierre, le 16 août 2010

Gaspoz Arnaud

12 Bibliographie

[1] Organe de coordination cybersanté, e-health-suisse.ch. [Online]
http://www.e-health-suisse.ch/index.html?lang=fr

[2] Office fédéral de la communication, bakom.admin.ch. [Online]
http://www.bakom.admin.ch/themen/infosociety/01689/index.html?lang=fr

[3] IHE International. ihe.net. [Online]
http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf - Page 7

[4] IHE International. ihe.net. [Online]
http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf - Page 15

[5] Karima Bourquard. Gmsih.fr. [Online]
www.gmsih.fr/content/download/865/5334/file/JFR-sécurité.pdf

[6] Wikipédia.org. [Online] Visité le 18 Juin 2010
http://fr.wikipedia.org/wiki/Health_level_7

[7] Wikipédia.org. [Online] Visité le 16 Juin 2010
http://fr.wikipedia.org/wiki/Digital_imaging_and_commmunications_in_medicine

[8] IHE International. ihe.net. [Online]
http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf - Pages 67-69

[9] IHE International. ihe.net. [Online]
http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf - Pages 71-72

[10] Medicoordination.ch. [Online] Visité le 15 Juin 2010
http://www.medicoordination.ch/

[11] YouTube.com. [Online] Visité le 15 Juin 2010
http://www.youtube.com/watch?v=IYSlIfWvNvw

[12] Bruno Alves, Henning Müller, Michael Schumacher David Godel. MediCoordination.ch. [Online]
http://www.medicoordination.ch/documents/delivrables/d2.2/D2.2-Analyse-des-requies-et-cas-utilisateurs_v3.2_final.pdf

[13] MediCoordination.ch, [Online] Visité le 15 Juin 2010
http://www.medicoordination.ch/index.php?option=com_content&view=article&id=44&Itemid=57

[14] Bruno Alves, Henning Müller, Michael Schumacher David Godel. MediCoordination.ch. [Online]
http://www.medicoordination.ch/documents/delivrables/d4/D4_Prototype_v1.0_final.pdf - Page 8

[15] Sondra Renly, Matthew Davis, Jesse Pangburn, Sarah Knoop, OpenHealthTools.org. [Online] Visité le 16 Juin 2010 *https://iheprofiles.projects.openhealthtools.org/wiki/*

[16] Matthew Davis, OpenHealthTools.org. [Online] *https://iheprofiles.projects.openhealthtools.org/wiki/Bridge/Overview*

[17] Microsoft.com. [Online] Visité le 16 Juin 2010 *http://download.microsoft.com/download/1/9/4/19404de3-cce6-4d4a-bfa5-8302ce4bb811/IHE%20XDS%20b%20Developer%20Whitepaper%205-2008.pdf*

[18] Office fédéral de la santé publique (OFSP), bag.admin.ch. [Online] Visité le 21 Juin 2010 *http://www.bag.admin.ch/themen/gesundheitspolitik/10357/10359/index.html?lang=fr*

[19] Organe de coordination Confédération-cantons. e-health-suisse.ch. [Online] *http://www.e-health-suisse.ch/umsetzung/00146/00148/index.html?lang=fr&download=NHzLpZig7t,lnp6I0NTU042l2Z6ln1ae2IZn4Z2qZpnO2Yuq2Z6gpJCDdHt4g2ym162dpYbUzd,Gpd6emK2Oz9aGodetmqaN19XI2IdvoaCVZ,s-*

[20] "Réunion du 27.05.2010 au RSV de Sion,".

[21] MediCoordination.ch. [Online] *http://www.medicoordination.ch/documents/delivrables/d4/D4_Prototype_v1.0_final.pdf* - Pages 44-46

[22] Wikipédia.org. [Online] Visité le 14 Juillet 2010 *http://fr.wikipedia.org/wiki/RBAC*

[23] Wikipédia.org. [Online] Visité le 13 Juillet 2010 *http://en.wikipedia.org/wiki/SAML*

[24] Wikipédia.org. [Online] Visité le 13 Juillet 2010 *http://en.wikipedia.org/wiki/XACML*

[25] Mark O'Neill, *Web Services Security*, Osborne/McGraw-Hill, Ed., 2003, Page 129.

[26] Wikipédia.org. [Online] Visité le 09 Juillet 2010 *http://fr.wikipedia.org/wiki/SOAP*

[27] Wikipédia.org. [Online] Visité le 09 Juillet 2010 *http://fr.wikipedia.org/wiki/Transport_Layer_Security*

[28] Mark O'neill, *Web Service Security*, Osborne/McGraw-Hill, Ed., 2003, Page 42.

[29] Wikipédia.org. [Online] Visité le 09 Juillet 2010 *http://fr.wikipedia.org/wiki/Open_Systems_Interconnection*

[30] Wikipédia.org. [Online] Visité le 09 Juillet 2010 *http://fr.wikipedia.org/wiki/WS-Security*

[31] Mark O'Neill, *Web Services Security*, Osborne/McGraw-Hill, Ed., 2003, Page 53.

[32] Mark O'Neil, *Web Services Security*, Osborne/McGraw-Hill, Ed., 2003, Page 52.

[33] Mark O'Neill, *Web Services Security*, Osborne/McGraw-Hill, Ed., 2003, Page 169.

[34] Wikipédia.org. [Online] Visité le 12 Juillet 2010 *http://en.wikipedia.org/wiki/WS-Security*

[35] Mark O'Neill, *Web Services Security*, Osborne/McGraw-Hill, Ed., 2003, Page 51.

[36] IHE International. ihe.net. [Online] *http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf* - Pages 163-165

[37] IHE International. ihe.net. [Online] *http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf* - Pages 20

[38] IHE International. ihe.net. [Online] *http://www.ihe.net/Presentations/upload/PIX_PDQ_RHIO_webinar_2005-08-18.ppt*

[39] IHE Internationale. ihe.net. [Online] *http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf* - Pages 102-106

[40] IHE International. ihe.net. [Online] *http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol2b_FT_2009-08-10.pdf* - page 92

[41] IHE International. ihe.net. [Online] *http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf* - Pages 54-55

[42] IHE International. ihe.net. [Online] *http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol2a_FT_2009-08-10-2.pdf* - Pages 178-181

[43] IHE International, ihe.net. [Online] Visité le 06 Juillet 2010 *http://www.ihe.net/Connectathon/*

[44] Wikipédia.org. [Online] Visité le 07 Juillet 2010 *http://fr.wikipedia.org/wiki/Subversion_%28logiciel%29*

[45] Wikipédia.org. [Online] Visité le 07 Juillet 2010 *http://fr.wikipedia.org/wiki/Apache_Maven*

[46] Wikipédia.org. [Online] Visité le 07 Juillet 2010 *http://fr.wikipedia.org/wiki/PostgreSQL*

[47] Ntp.org. [Online] Visité le 19 Juillet 2010
http://www.eecis.udel.edu/~mills/ntp/html/accopt.html

[48] Ntp.org. [Online] Visité le 19 Juillet 2010
http://www.eecis.udel.edu/~mills/ntp/html/authopt.html

[49] Gentoo-wiki.com. [Online] Visité le 20 Juillet 2010
http://fr.gentoo-wiki.com/wiki/NTP

[50] Linuxhomenetworking.com. [Online] Visité le 20 Juillet 2010
http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch24_:_The_NTP_Server

[51] Freebsd.org. [Online] Visité le 20 Juillet 2010
http://docs.freebsd.org/doc/7.1-RELEASE/usr/share/doc/fr/books/handbook/network-ntp.html

[52] Postgresqlfr.org. [Online] Visité le 22 Juillet 2010
http://docs.postgresqlfr.org/8.4/client-authentication.html

[53] Paul Abbott, Ibm.com. [Online] Visité le 29 Juillet 2010
http://www.ibm.com/developerworks/java/library/j-certgen/

[54] Wikipédia.org. [Online] Visité le 29 Juillet 2010
http://fr.wikipedia.org/wiki/Public_Key_Cryptographic_Standards

[55] Microsoft.com. [Online] Visité le 29 Juillet 2010
http://msdn.microsoft.com/fr-fr/library/ms733791.aspx

[56] Openhealthtools.org. [Online] Visité le 04 Août 2010
https://iheprofiles.projects.openhealthtools.org/wiki/Bridge/Operations/SessionContext

13 Abréviations

MC	MediCoordination
XDS	Cross-Enterprise Document Sharing
PDF	Portable Document Format
ITI TF	IHE IT Infrastructure Technical Framework
HL7	Health Level 7
ISO	Organisation internationale de normalisation
DICOM	Digital Imaging and COmmunications in Medicine
CDA	Clinical Document Architecture
XML	Extensible Markup Language
URI	Uniform Resource Identifier
ATNA	Audit Trail and Node Authentication
PDQ	Patient Demographics Query
PIX	Patient Identifier Cross-Referencing
PAM	Patient Demographics Source
XCA	Cross Community Access
XUA	Cross-Enterprise User Assertion
IIP	IHE Integration Profile
OHT	OpenHealthTools
FTP	File Transfer Protocol
SOAP	Simple Object Access Protocol
EAN	European Article Number
LPD	Loi fédérale sur la protection des données
OLPD	Ordonnance relative à la loi fédérale sur la protection des données
TLS	Transport Layer Security
RSV	Réseau Santé Valais
GAC	Global Assembly Cache
RBAC	Role-Based Access Control

SOA	Service Oriented Architecture
SAML	Security Assertion Markup Language
SSO	Single Sign-On
XACML	eXtensible Access Control Markup Language
PEP	Policy enforcement point
PDP	Policiy decision point
PRP	Policy retrieval point
PAP	Policy administration point
PIP	Policy information point
RPC	Remote Procedure Call
SSL	Secure Sockets Layer
OSI	Open Systems Interconnection
WSS	WS-Security
W3C	World Wide Web Consortium
IETF	Internet Engineering Task Force
ASN	Abstract Syntax Notation
BPPC	Basic Patient Privacy Consents
HPC	Health Professional Card
EUA	Enterprise User Authentication
OWASP	Open Web Application Security Project
CCOW	Clinical Context Object Workgroup
NTP	Network Time Protocol
HTML	HyperText Markup Language
CA	Certificate Authority
BSD	Berkeley Software Distribution
UDP	User Datagram Protocol
DBA	Database Administrator
RHIO	Regional Health Information Organization
DN	Distinguished Name
PKCS	Public Key Cryptographic Standards

14 Table des illustrations

14.1 Images

14.2 Tableaux

15 Annexes

15.1 Cahier des charges

Cahier des charges

Projet : Exchange of Medical Informations

HES-SO Valais // Wallis 2010 - Travail de Bachelor
Gaspoz Arnaud

Introduction

Le domaine de la santé est actuellement en train de passer d'une méthode basée sur le papier à une méthode électronique. La Confédération suisse porte une grande attention à la cybersanté (eHealth) et vient de publier une nouvelle stratégie. Le but est de numériser les informations médicales, puis de les rendre accessibles immédiatement.

Ce travail de bachelor sera effectué dans le contexte du projet MediCoordination[1] (MC) dirigé par l'institut Informatique de gestion de Sierre. Un prototype a déjà été implémenté et permet l'échange de données médicales entre hôpitaux et médecins généralistes. L'implémentation s'est basée sur les profils IHE[2] (Integrating the Healthcare Entreprise).

Table des matières

[1] http://www.medicoordination.ch/
[2] http://www.ihe.net/

31.05.2010 1

Objectifs

Voici les différents objectifs à réaliser :

- Comprendre l'architecture actuelle
- Elaborer des audits de sécurité
- Analyse des profils IHE pouvant répondre à certains problèmes de sécurité
- Conception d'un ou plusieurs aspects décrits dans l'audit
- Proposer une implémentation

Comprendre l'architecture actuelle

La première phase du travail de bachelor consistera à comprendre l'architecture du prototype de MediCoordination qui se base sur le profil Cross Entreprise Document Sharing (XDS) d'IHE.

Ainsi, la lecture du document officiel des profils IHE permettra de saisir leur but et donc d'avoir une vue globale de leur utilité. Dans un second temps, le profil IHE XDS sera examiné avec plus de détails.

Plusieurs outils ont été utilisés pour la conception de ce prototype, dont iheprofiles d'OpenHealthTools[3] et IHE Integration Profile[4] de Microsoft.

Sur le portail du projet MediCoordination, différents documents sont à disposition permettant une vision détaillée de l'implémentation. Il est important d'assimiler les subtilités du prototype afin de pouvoir porter un jugement sur le travail accompli.

Un soin particulier sera mis à l'analyse des documents émis par l'organe de coordination « Cybersanté Suisse ».

Elaborer des audits de sécurité

En deuxième phase, un regard doit être porté sur les aspects de sécurité du prototype, en particulier les aspects de confidentialité et de protection des données, ainsi qu'éventuellement l'ajout de rôles pour les personnes médicales et les patients.

Un point important concerne les exigences liées au lieu de l'implémentation. Le prototype étant installé dans le réseau d'un hôpital, il faut prendre en compte les remarques du responsable de la sécurité informatique. Une séance a été mise sur pied le 27 mai 2010 au RSV de Sion. Quelques points y ont déjà été relevés et un document plus complet sera transmis dans quelques jours.

Une analyse personnelle du prototype sera également effectuée. Le travail accompli dans ce domaine sera mis en évidence et les points sensibles seront soulevés.

Analyse des profils IHE pouvant répondre à certains problèmes de sécurité

Après avoir relevé quelques failles possibles dans le prototype, la recherche de solutions est absolument obligatoire dans un domaine tel que l'échange de données médicales.

[3] http://www.openhealthtools.org/
[4] http://ihe.codeplex.com/

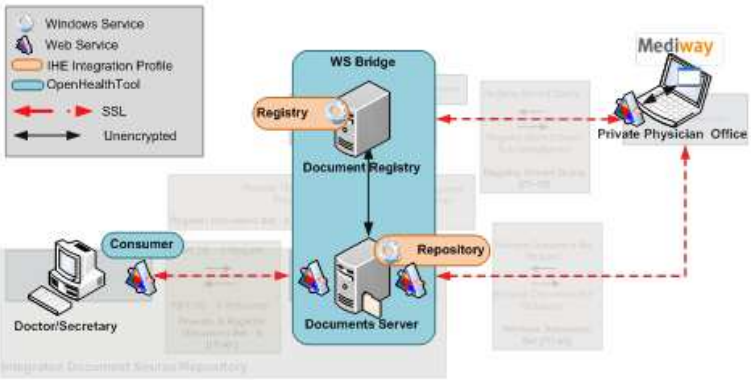
Dans la documentation technique des profils IHE, des recommandations liées à la sécurité sont exposées. Une comparaison de celles-ci avec les points relevés dans la seconde partie est donc importante pour rendre l'application plus sûre et également se rapprocher le plus possible des recommandations d'IHE.

Plusieurs profils IHE seront donc proposés afin d'améliorer certains aspects de la sécurité du prototype.

Il faudra ensuite faire une recherche des différents outils disponibles pour implémenter des profils dans notre implémentation et déterminer s'ils sont compatibles avec ceux déjà en place, tels qu'OpenHealthTools.

Conception d'un ou plusieurs aspects décrits dans l'audit

Le but de cette partie est de modifier l'implémentation actuelle en intégrant le ou les profils choisis dans l'audit de sécurité :



Un nouveau schéma de l'architecture contiendra les extensions en rapport aux profils IHE sélectionnés.

Proposer une implémentation

L'implémentation doit au moins porter sur un profil IHE. Les outils trouvés dans la phase d'analyse seront installés et configurés dans le prototype actuel. Cette implémentation se fera dans un environnement de test.

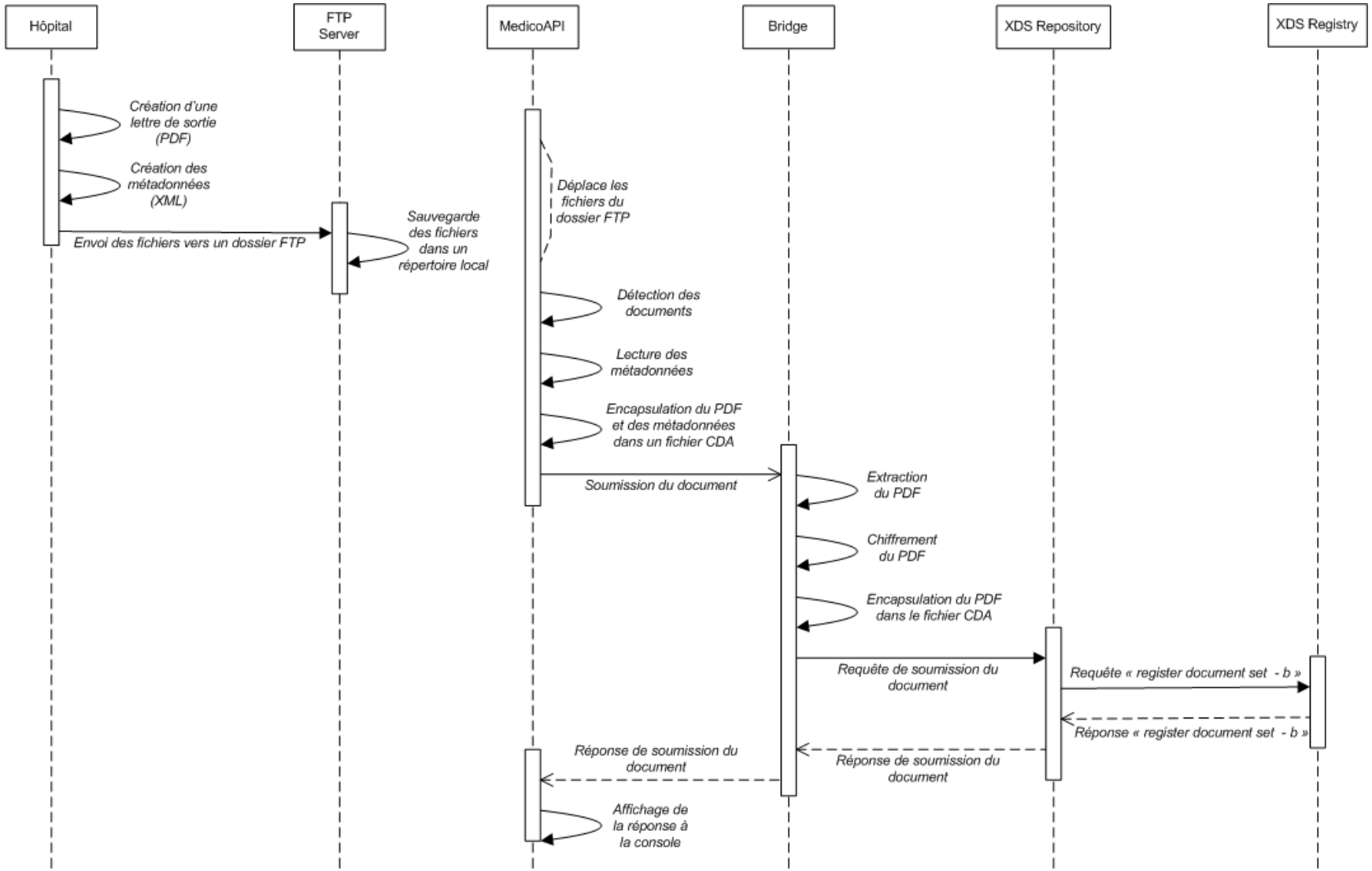
Signature

Par leurs signatures, les soussignés déclarent avoir lu et approuvé l'entier du document.

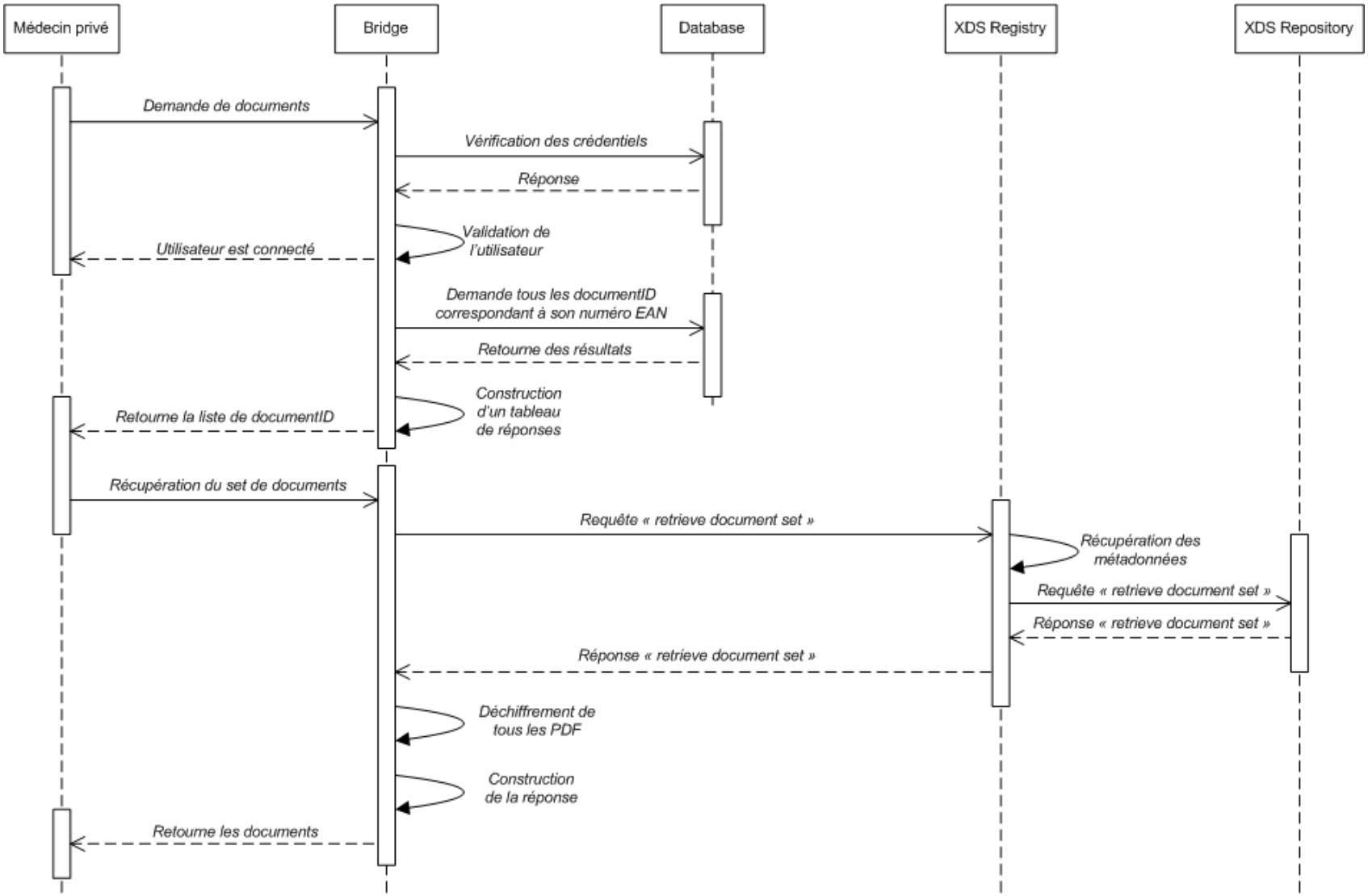
	Date	Signature	Nom
Pour le client :	.....	.....	Michael Schumacher
Exécutant :	.....	.....	Arnaud Gaspoz

31.05.2010 3

15.2 Processus de soumission d'un document



15.3 Processus de récupération d'un document



15.4 Planifications initiale et finale